

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

**THIS PAGE BLANK (USPTO)**

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
8 March 2001 (08.03.2001)

PCT

(10) International Publication Number  
**WO 01/16792 A1**

(51) International Patent Classification<sup>7</sup>: **G06F 17/24,**  
17/30

G. [US/US]; 2320 Blue Spruce Lane, Aurora, IL 60504 (US). **SHIVELY, J., Thomas** [US/US]; 112 N. Washington Street, Hinsdale, IL 60521 (US).

(21) International Application Number: PCT/US99/20203

(22) International Filing Date:  
1 September 1999 (01.09.1999)

(74) Agent: **MCCRACKEN, William, E.**; Marshall, O'Toole, Gerstein, Murray & Borun, 233 S. Wacker Drive, 6300 Sears Tower, Chicago, IL 60606 (US).

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US): **R.R. DONNELLEY & SONS COMPANY** [US/US]; 3075 Highland Parkway, Downers Grove, IL 60515-1261 (US).

(81) Designated States (national): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW.

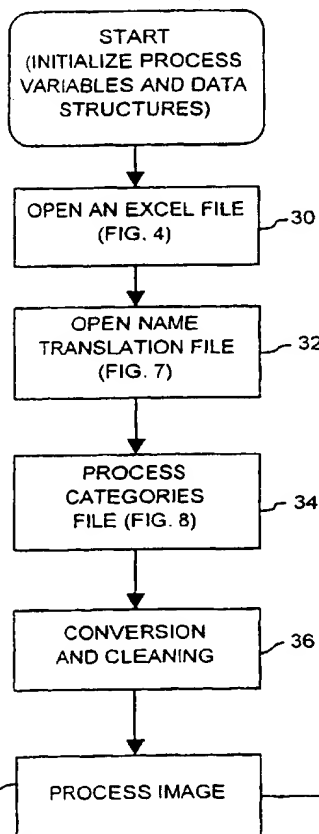
(72) Inventors; and

(75) Inventors/Applicants (for US only): **DREYER, Mark,**

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT,

[Continued on next page]

(54) Title: AUTOMATED CONVERSION OF PRINT-READY DOCUMENTS FOR DISPLAY



(57) Abstract: A system for converting a page layout file into a database for use by a display program which converts the database for display in a medium opens the page layout file, converts a portion of the page layout file into data compatible with the medium and assembles the data into the database.



WO 01/16792 A1



BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,  
NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA,  
GN, GW, ML, MR, NE, SN, TD, TG).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**Published:**

— *With international search report.*

AUTOMATED CONVERSION OF PRINT-READY  
DOCUMENTS FOR DISPLAY

Field of the Invention

5           The present invention relates generally to conversion systems and methods, and more particularly to an automated apparatus and method for converting information from a print-ready document into a format useful for display.

10       Background Art

          Printed documents are often assembled into books, such as catalogs, brochures, manuals, and the like. Such documents may have diverse content including text, images and line art positioned in a myriad of ways. Through the use of page make-up software, a printed document can be designed using a  
15       computer. In the specific example of printed catalogs, the page make-up software is used to design the pages of the catalog such that they include, for example, item names, descriptions, images, price, etc...

20       In addition to print media, it is often desirable to utilize other types of media to communicate information. For example, it has been found desirable to create on-line

versions of print catalogs. In one application, information from a print catalog page layout file is used to create a catalog web page accessible via an on-line computer network. This on-line catalog lists properties of catalog items, such as description, price, SKU number and the like. Typically, the method of creating on-line catalogs begins with an operator opening an existing page make-up file, such as a QuarkXpress® file. The page make-up file contains detailed information for each of a number of products offered for sale, such as name, description, price and catalog number, and further may include an image of the product. In addition to this detailed information, the page make-up files usually contain a number of embedded characters which serve numerous functions, such as denoting non-ASCII characters for printing or carriage returns and the like, or these embedded characters may comprise control codes. Such characters can cause unpredictable and/or undesirable results when present in the coding for the on-line web pages. Accordingly, these characters must be stripped from the page make-up files, together with undesirable characters, such as double spaces. Also, in some cases, different codes must be inserted into the

coding for the web pages to permit proper display of characters.

Conventionally, the embedded characters were removed by an operator. Specifically, with the page make-up file opened, the operator manually selected text and pasted the selected text into an intermediate file created by an application which automatically stripped certain codes during the ensuing conversion process, such as Microsoft Word®. However, since neither Word® nor any other software automatically removes and/or corrects all undesirable characters, the operator must still analyze the page make-up file(s) and remove and/or correct all remaining undesirable characters. Once this has been accomplished, the information must be manually parsed to permit the different types of information (such as price, description, SKU number, color, etc...) to be stored in different fields of a record of a database file, such as a Microsoft Excel® file. In addition, if an image of the product is to be displayed, an image file name must be stored in a further field of the database record. The resulting database file is then used to dynamically produce web pages.

While the foregoing has been effective to permit catalog web pages to be produced, the number of manual steps required multiplied by the number of products in a typical catalog results in the expenditure of a large number of man-hours to allow even a modest catalog to be displayed on-line. In addition to being a relatively expensive undertaking, there is a substantial chance that errors will be introduced. Further, the amount of time required to effect the conversion results in a long cycle time before the on-line pages are available.

10



Summary of the Invention

In accordance with one aspect of the present invention, a system for converting a page layout file into a database for use by a display program which converts the database for display in a medium comprises means for opening the page layout file, means responsive to selection of a portion of the page layout file for converting the portion into data compatible with the medium and means responsive to the converting means for assembling the data into the database.

In accordance with an alternative aspect of the present invention, a software system for converting a page layout file into a database for use by a display program wherein the display program converts the database for display in a certain medium includes a computer-readable medium and a software program stored in the computer-readable medium. The software program includes a first routine that opens the page layout file, a second routine responsive to selection of a portion of the page layout file and that converts the portion into data compatible with the certain medium and a third routine responsive to the second routine and that assembles the data into the database.

In accordance with a further aspect of the present invention, a software system for converting a page layout file representing a catalog page into a database for use by a program which converts the database into web pages encoded in Internet web format wherein the catalog page includes a text portion and an image portion wherein the text and image portions relate to a catalog object includes a computer-readable medium and a software program stored in the computer-readable medium. The software program includes a first routine that opens the page layout file, a second routine responsive to selection of one of the text portion and the image portion and that converts the portion into data compatible with the web format and a third routine responsive to the second routine and that assembles the data into the database and relates such data to the object.

In accordance with yet another aspect of the present invention, a system for converting a page layout file for display in a medium includes means for opening the page layout file, means responsive to selection of the portion of the page layout file for converting the portion into data compatible with the medium and means responsive to the converting means

for assembling the data into a database. The system further includes a display program which converts the database into page description files for display in the medium.

In accordance with a still further aspect of the present invention, a method of converting a page layout file for display in a medium includes the steps of opening the page layout file, converting the portion into data compatible with the medium in response to selection of a portion of the page layout file, assembling the data into a database and using a display program to convert the database into page description files for display in the medium.

Other aspects and advantages of the present invention will become apparent upon consideration of the following drawings and detailed description.

#### Brief Description of the Drawings

FIG. 1 is a sample page of a catalog as represented by a page make-up program, such as QuarkXpress®;

FIG. 2 illustrates a main dialog box which is displayed to an operator during the process of converting page make-up

files to page description files according to the present invention;

FIG. 3 is a high-level flowchart of a portion of the process according to the present invention;

5        FIG. 4 is a flowchart of programming executed by the block 30 of Fig. 3;

FIGS. 5A-5C, when joined side-by-side with Fig. 5A on the left, Fig. 5B in the middle and FIG. 5C on the right, is an illustration of the product table noted in FIG. 4;

10       FIG. 6 is an illustration of the SKU table noted in FIG. 4;

FIG. 7 is a flowchart of programming executed by the block 32 of FIG. 3;

15       FIG. 8 is a flowchart of programming executed by the block 34 of FIG. 3;

FIG. 9 is a flowchart of programming executed by the block 36 of FIG. 3;

FIG. 10 is an illustration of the dialog box of FIG. 2 with sample information entered therein;

20       FIG. 11 is a flowchart of programming executed by the block 154 of FIG. 9;

FIGS. 12 and 13 are illustrations of dialog boxes which are presented to an operator during execution of the programming of FIG. 11;

FIG. 14 is a flowchart of programming executed by the block 198 of FIG. 11;

FIG. 15 is a flowchart of programming executed by the block 158 of FIG. 9;

FIG. 16 is an illustration of a dialog box which is presented to an operator during execution of the programming of FIG. 15;

FIG. 17 is a flowchart of programming executed by the block 162 of FIG. 9;

FIG. 18 is a flowchart illustrating the a portion of the programming executed by the block 166 of FIG. 9 when the "Save" button of FIG. 10 is selected;

FIGS. 19A-19C, when joined side-by-side with FIG. 19A on the left, FIG. 19B in the middle and FIG. 19C on the right, is an illustration of the product table with sample data stored therein;

FIGS. 20A-20C, when joined along similarly lettered lines, together represent programming executed by the block 38 of FIG. 3;

FIG. 21 is an illustration of a dialog box which is presented to an operator during execution of the programming of FIGS. 20A-20C;

FIG. 22 is a flowchart of programming executed by the blocks 432 and 424 of FIGS. 20A and 20C, respectively;

FIG. 23 is a block diagram of a computer system which may be used to implement the present invention; and

FIG. 24 is a block diagram of a process of converting print-ready documents to web pages according to the present invention.

#### Description of The Preferred Embodiment

The present invention is described below in the context of converting the contents of QuarkXPress® print-ready documents into one or more Microsoft Excel® database tables and image(s) converted into a web-ready (i.e., Internet) format using Adobe Photoshop®. The contents of the database tables can then be converted to a web format and the resulting

text coding and web-ready image(s) can then be easily formatted by a display program known as Marketplace (Version 2) licensed by Multimedia Live of Petaluma, California, to dynamically generate web pages for display by a server (not shown) on the Internet (FIG. 24 illustrates this process). However, it should be apparent to one of ordinary skill in the art that the conversion process and utility described herein may be used in other applications or with other page layout programs, image manipulation programs and/or page development programs and still be within the scope of the present invention. For example, page make-up files produced by any page make-up program could be converted into database files and converted image files of any type and the database files and the converted image files could be used to develop page description files (again, of any type) for display on or in any target or output medium, such as a computer monitor, ink or toner on paper, dye in film, or the like. Also, while the present invention is described in the context of converting the pages of a print-ready catalog into web pages, it should be understood that other printed matter may instead be converted for display.

Initially, one or more page make-up files (also referred to as layout files) are created using a computer system running a page make-up (or layout) program, such as QuarkXpress®. The computer system may be like that shown in FIG. 23 including a computer 25 having internal memory 26, input devices 27, such as a keyboard and mouse, a monitor 28 and, optionally, a connection to a network 29. The page make-up file(s) represent a book which has or will be produced, such as a catalog or other printed matter. Generally, these page layout file(s) can be stored on a computer-readable memory such as a hard drive or other storage device or the page layout file(s) may be stored and accessed remotely through the network 29. FIG. 1 illustrates a sample page from a seed catalog as displayed by QuarkXpress®. The sample page includes three catalog items (alternatively termed objects herein) referred to by the large text "Sugar Crunch," "Watercolor Memories" and "Super Tasty." All catalog items contain a text portion and a high resolution image. The text portion for each catalog item contains detailed information including item name, description, catalog number or SKU number, style description, and price. Other information and variations on this detailed



information may be provided. Furthermore, not every catalog item need have the same types or levels of detailed information.

The text portions of the page layout file are untagged and unstructured, i.e., they are not stored as separate character strings based on the contents thereof but rather as a single group of characters. Thus, while the text "Sugar Crunch" is the name of one of the catalog items, "B-54031" is the item number and "\$2.95" is the price for 30 seeds, all of this text is stored as one complete group of characters in the page layout file, with embedded characters indicating special symbols, fractions, end-of-line, etc... Furthermore, these text items are stored in the page layout file without relational linking among the text items, e.g., the price string for "Sugar Crunch" is not relationally linked to the item number string even though they both relate to the same catalog item or object.

According to the present invention, an automated conversion process converts the text portions of the page layout file into a database and converts the image(s) into a proper format for generation of web pages or another type of

visual display. A main dialog box for the conversion program is shown in FIG. 2 and is opened by a user operating a computer system like that shown in FIG. 23. The page layout file is also opened using the native application that created the page layout file (in the illustrated example,

QuarkXpress®). The dialog box includes fields for substantially all of the text items relating to a given catalog item. Each dialog box field contains a label indicating the field name (i.e., "Item Number," "Item Name,"

"Headline," "Tn Description," (for thumbnail description) "Image Name," "Keywords," "Description," "Detail Image(s)," "Category Information" and "SKU Information") and a text box into which converted text from the page layout file or information generated by the user is placed and stored. The

dialog box fields preferably coincide with the contents of the catalog items in the page layout file, and therefore may vary from catalog to catalog. In the present example, the fields identified by the labels "Item number," "Item name,"

"Headline," and "Description" are referred to as conversion fields because text from the page layout file is automatically converted to a desired format and cleaned (i.e., undesired

codes are removed and other codes are inserted to cause proper display of text) by a conversion program when the text is placed into such fields. This conversion and cleaning can be performed in a manner unique to each conversion field or some or all of the conversion fields can employ the same conversion and cleaning process. The remaining fields are referred to as user fields because the information stored therein is provided by the user (except for some fields in the "SKU Information" area), as described in detail below.

10 A process in accordance with the present invention is shown in the generalized flow diagram of FIG. 3. The steps of FIG. 3 may be executed by a user using the computer system of FIG. 23 after the dialog box of FIG. 3 and the page layout file are opened. The steps of FIG. 3 and the flowcharts  
15 following FIG. 3 may be partially or fully implemented by software programmed in a desired programming language, such as AppleScript® by Apple Computer, Inc. At least a portion of the programming described herein is set forth in the AppleScript® coding included in the Appendix attached hereto.  
20 After initialization of process variables and data structures, a block 30 opens a database file using a native application,

such as Excel®. The database file comprises a template which stores converted and cleaned data from the conversion fields of the dialog box of FIG. 2. Next, a name translation file is opened by a block 32. The name translation file stores the  
5 names of images in the page layout file that are to be processed. Thereafter, a categories file, which contains previously stored information listing possible categorizations of a particular catalog item, is opened and processed by a block 34, again using a native application, such as Excel®.

10 In summary, the blocks 30, 32 and 34 open files that will be accessible during use of the dialog box shown in FIG. 2.

Next, at a block 36, the user can select text displayed by the page make-up program by highlighting such text using the mouse and clicking on one of the conversion fields of the dialog box  
15 of FIG. 2. This action causes the block 36 to run the conversion program to convert and clean the text selected from the page layout file. The converted and cleaned text is also displayed in the selected conversion field.

After the block 36, a block 38 runs an image conversion  
20 program which converts any image(s) into a format suitable for display.

Though the process steps of FIG. 3 are shown in a particular order, one of ordinary skill in the art will appreciate that the first three steps 30, 32 and 34 may be performed in any order and that the steps 36 and 38 may be undertaken independently. Furthermore, it should be appreciated that any of the steps may be initiated from the dialog box of FIG. 2. By way of example, the categories file could be opened from a specified directory by clicking on one of the boxes of the category information field of the dialog box of FIG. 2.

A detailed flow diagram of the programming executed by the block 30 of FIG. 3 is shown in FIG. 4. A block 62 displays a dialog box prompting the user to open a new or existing Excel<sup>®</sup> database file. Control pauses at a block 64 until a database file is selected. Each database file is preferably organized into multiple tables to separate different types of information. For example, a database file may contain a product table 66, a SKU table 68, an alternate image table 70, etc... The database file may also include scratch tables which temporarily store data. Each table comprises a plurality of records, and each record includes a

number of fields each organized under a field heading. A sample empty product table is shown in FIGS. 5A, 5B and 5C. The product table has item number, item name, keywords, category, manufacturer name, manufacturer logo, main category 1 and 2, sub category 1 and 2, detail category 1 and 2, headline, thumbnail description, thumbnail image, product description, product page image, publish, default price, and tax exempt headings. These headings are exemplary in nature and may be replaced by other headings depending upon the particular catalog or other printed matter that is being converted for display. Other field headings may also be used depending upon whether the page layout file contains more information to be characterized and displayed. Preferably, there is at least one product table heading for each of the conversion fields of the dialog box of FIG. 2.

A detailed view of the SKU table is shown in FIG. 6. The SKU table is used to track items for availability and promotion. The SKU table may have, for example, a heading for promotion name, promotion description, promotion rank, start and end dates of promotion, etc.

The alternative image table is used to track "detail images." A detail image is an image derived from the high resolution image of an object. For example, a detail image may be a cropped and enlarged portion of a high resolution image.

Referring again to FIG. 4, once the user has selected a database file, a block 66 opens the tables of the file and a block 68 searches for the next available open (or blank) row in each table. Because these rows may not be in the same location in the various tables, the locations of these next available rows are noted and stored by the block 68 so that all of the information for a given object that is populated into the database records will be linked together. After the next open row for each table is found, the rows are marked for data entry by the block 68 and thereafter the database file selection process ends.

FIG. 7 illustrates the programming executed by the block 32 of FIG. 3. The name translation file is a file which maps the names of the high resolution image files in the page layout file to the names of images that will be created during the image processing of the block 38 of FIG. 3. As discussed

below, the high-resolution images of the page layout file, which vary in size and shape, may be converted in an automated fashion into images that can be displayed in or on the target medium (e.g., the Internet) including a thumbnail image and/or  
5 a cropped image. The name translation file stores the names of all images that are noted in the fields labeled "Detail Image(s)" of the dialog box of FIG. 2.

A block 90 displays a dialog box (not shown) prompting the user to select opening of either an existing or a new name  
10 translation file. If the user selects that an existing file is to be opened, which would be done to continue inputting image names from a previously saved catalog conversion, a block 92 displays a dialog box (also not shown) which presents all existing file names for selection by the user. Once the  
15 user indicates selection of a particular file, a block 94 opens the selected file and determines the next empty row in the table. Control then passes to the block 34 of FIG. 3. If the operator opens a new file, a block 96 displays a dialog box (not shown) which prompts the user to enter a file name  
20 and a block 98 opens the file. A block 100 then sets the file length to a null value so that all data that may be in the



file are cleared. Control then passes to the block 34 of FIG. 3.

The block 34 of FIG. 3 processes a category file created by the same user or a different user. Generally, an object can be categorized to permit the objects to be searched at a later time. Each object can be categorized at three levels under the "Main," "Sub," and "Detail" headings of the fields labeled "Category Information" of the dialog box of FIG. 2. In order to facilitate this categorization, the category file is processed to provide the possible listings for a main category, a sub-category and a detail category. A sample format for a category file is listed in TABLE 1 below:

TABLE 1

	<u>C</u>	<u>S</u>	<u>D</u>	<u>STR</u>
	1	0	0	Lawn & Garden
	1	1	0	Flowers
5	1	1	1	Geraniums
	1	1	2	Marigolds
	1	1	3	Sunflowers
	1	1	4	Pansy
	1	1	5	Phlox
10	1	1	6	Iris
	1	1	7	Daisy
	1	2	0	Vegetables
	1	2	1	Tomatoes
	1	2	2	Squash
15	1	2	3	Gourds
	1	2	4	Melons
	1	2	5	Beans
	1	2	6	Cucumbers
	1	2	7	Swiss Chard
20	1	2	8	Onions
	1	2	9	Eggplant
	2	0	0	Toys & Games

25       TABLE 1 includes a variable C identifying a number of possible main categories, a variable S identifying a number of possible sub-categories and a variable D identifying a number of possible detail categories. The fourth column of TABLE 1 contains a string STR identifying the name or title of the

30       corresponding categorization. For example, the first entry in the table includes the string "Lawn & Garden." Because this entry has a zero value for the sub-category variable S and the detail variable D, this string is a main category title.

Similarly, "Toys & Games" is a main category title, as indicated by the stored values of C=2, S=0, and D=0. There are two sub-categories under the "Lawn & Garden" main category identified as "Flowers" and "Vegetables," as indicated by the zero value for the variable D. Under the "Flowers" sub-category, there are numerous detail categories, such as "Geraniums," "Marigolds," "Sunflowers," "Pansies" and "Phlox." Detail categories such as "Tomatoes," "Squash," "Beans" and "Cucumbers" are arranged under the "Vegetables" sub-category (which, in this case, results in an incorrect categorization because tomatoes are a fruit and beans are legumes). The category file contains as many main categories, sub-categories and detail categories as are desired by the user or called for by the design of the web pages. Preferably, there are at least enough categories to adequately describe all of the objects in the print-ready file.

FIG. 8 illustrates the programming executed by the block 34 of FIG. 3. A block 120 displays a dialog box which prompts the user to select a categories file for processing. Once a file is selected by the user, the categories file is opened by a block 122 and a block 124 checks to determine whether all

records in the categories file have been processed. If not, a block 126 assigns the values of the next record in the file to variables C, S, D, and STR. A block 128 then checks to determine whether the value of S is equal to zero. If this is the case, a block 130 stores the string STR in memory as a main category. If the block 128 determines that the value of S is not equal to zero, a block 132 determines whether the value of the variable D is equal to zero. If this is true, a block 134 stores the string STR in memory as a sub-category under the main category identified by the values C=X, S=0 and D=0, where X is the value of C for the current record.

If the block 132 determines that the value of D is nonzero, a block 136 stores the string STR as a detail category in memory under the subcategory identified by the values C=X, S=Y and D=0, where Y is the value of S for the current record.

It should be evident from the foregoing that the programming of FIG. 8 is dependent upon the format of the categories file and that such programming may change if the categories file format is different from that set forth above.

With the pre-conversion steps complete, the block 36 of FIG. 3 performs the conversion and cleaning process. A detailed description of the programming executed by the block 36 is presented in FIG. 9.

Initially, a block 150 displays the main dialog box of FIG. 2. The user then can highlight a portion of the text on the page layout file and click on a corresponding one of the conversion fields. Alternatively, the user can click on any other field or button without first highlighting text in the page layout file. Once an action is taken by the user, a series of blocks 152, 156, 160 and 164 determine whether the user has clicked on a field in the "SKU Information," "Category Information" or "Detail Image(s)" areas or on the "Image Name" field or on one of the buttons at the bottom of the dialog box. If any of these conditions is found to be true, control passes to one of blocks 154, 158, 162 or 166. The programming executed by blocks 154, 158 and 162 is shown in detail in FIGS. 11, 15 and 17, respectively. In the case of the block 166, clicking on the button labeled "Reset Excel" causes the database file opened by the block 30 of FIG. 3 to be reset, i.e., the contents of the database file are deleted

from the table. Clicking on the button "Save Excel" causes the information in the database file to be saved. Clicking on the button "Edit on SKU WS" causes the Excel® program to be displayed with the SKU table opened to permit direct editing thereof. Clicking on the button "Save SKU WS" causes the SKU table to be saved. If the user clicks on the "Reset" button, the contents of all the fields of the dialog box of FIG. 3 are erased. If the user clicks on the "Save" button, the data in the various fields of the dialog box are saved to the appropriate files. Clicking on the "Done" button causes the information in the fields of the dialog box to be saved to the appropriate files and closure of the dialog box.

If none of the conditions checked by the blocks 152, 156, 160 or 164 is found to be true, then it has been determined that the user has clicked on one of the conversion fields, and hence a block 168 automatically converts and cleans the text that was highlighted by the user. The highlighted information is filtered to remove and/or replace characters which, if unaltered, would not display properly in some or all web browsers. Control from each of the blocks 154, 158, 162, 166 and 168 returns to the block 150.

Preferably, all of the information for a specific conversion object is characterized (by highlighting specific text and clicking on the appropriate conversion field) and cleaned before converting information for the next conversion object. For example, referring again to FIG. 1, all of the information for the object "Sugar Crunch" is characterized and cleaned before processing is undertaken for the object "Watercolor Memories."

FIG. 10 illustrates the dialog box of FIG. 2 with information entered for the "Sugar Crunch" object. The text strings in the "Item Number," "Item Name," "Headline" and "Description" fields are separately entered into such fields, at which time the strings are converted and cleaned. The text strings in the "Tn Description," "Keywords" and "Detail Images" fields are entered by the user. The information in the "Category Information" fields is entered by successively clicking on the fields under the "Main," "Sub" and "Detail" headings, in turn causing selections for such fields to be displayed by a pop-down menu. The appropriate information may then be selected by the user by clicking on same.

Some of the information in the "SKU Information" field is entered like the information in the conversion fields (at which point such data are cleaned and converted) while the remaining information is manually entered by the user. Before  
5 describing this process further it is helpful to describe the ways in which an object is identified. In the illustrated example, each catalog object is identified by at least one item number which is stored in the "Item Number" conversion field. In addition to the item number, a catalog object is  
10 identified by at least one SKU number. Usually, an object is identified by more than one SKU number when the object may have differing attributes. For example, different SKU numbers may denote seeds which are for the same plant or vegetable except that the seeds have variations in style, colors, or  
15 sizes. Thus, seeds which grow into a blue version of a flower may have a different SKU number than seeds which grow into a red version of the same flower. The "SKU Information" area of the dialog box includes fields which allow object attributes and other information relating to an object to be specified.  
20 The fields include SKU, Style, Price, Color, Size, Sale Price ("Sale P"), Quantity Price ("Quan"), Quantity Needed ("Quan



N"), Thumbnail Price ("TP"), Weight ("W") and M. (When a "Y" is stored in the M field, the data in the respective row of the "SKU Information" table are stored in the SKU table as marked text, which may be in a different color than the remaining data in the SKU table. This facilitates location of such data so that the data can be further processed as desired.) Other fields may be used depending on user preference and/or catalog content.

The programming executed by the block 154 of FIG. 9 is shown in detail in FIG. 11. The programming is initiated by clicking on one of the "SKU Information" fields of the dialog box of FIG. 2. A block 190 then opens a dialog box as illustrated in FIG. 12. This dialog box contains fields for SKU Number, Style, Price, Color, Size, Sale Price, Quantity Price, Quantity Needed, and Weight, all of which have corresponding fields in the main dialog box of FIG. 2. Once the user clicks on a field or button of the dialog box of FIG. 12, a block 192 determines whether a button denoted "Combos" has been clicked on. If this is true, a block 194 displays the dialog box illustrated in FIG. 13 to allow the user to specify combinations of attributes (this operation is

discussed in greater detail hereinafter.) Control then returns to the block 190. If the block 192 determines that the "Combos" button has not been clicked on, a block 196 determines whether one of the text fields has been clicked on.

5 If this is true, a block 198 executes the routine executed in FIG. 14. This allows the user to enter SKU data through the use of the highlight text process described above with respect to the inputting of data into the conversion fields. Thus, if a page layout file has SKU information the user may highlight  
10 that information within the page layout file and click on the respective SKU conversion field to convert and clean the data in the same manner as described above and then store clean data into the SKU conversion fields.

If the block 196 determines that one of the text fields  
15 has not been clicked on, a block 200 determines whether an "OK" button has been clicked on. If this is the case, the data in the text boxes are transferred to the fields of the "SKU Information" area of the dialog box of FIG. 2 by a block 201. Control then passes to the block 150 of FIG. 9.

20 Otherwise, a block 202 executes programming which checks to determine whether a "Thumbnail Price" box, an "Apply" button,

a "Copies" field, a "Delete" button, a "Mark Item" box or a "Cancel" button has been clicked on. If the "Thumbnail Price" box has been clicked on, then a value is loaded into a field referred to as "Thumbnail Price" in the database. If the "Apply" button has been clicked on, then the values in the "Item Number," "SKU Number," "Size," "Sale Price," "Quantity Price," "Quantity Needed," "Weight" and "Thumbnail Price" fields are loaded into temporary memory for use during "Combo" processing, as noted hereinafter. If the "Copies" field has been clicked on, the user may specify that the information in text fields may be copied a specified number of times into the SKU table. If the "Delete" button has been clicked on, the corresponding record in the SKU table is deleted. When the "Mark Item" box has been clicked on, the entry in the SKU table corresponding to the information in the text boxes is marked as noted above. If the "Cancel" button has been clicked on, the data in text fields (if any) are deleted. Control from the block 202 then returns to the block 190.

Alternatively, if desired, SKU data may be automatically stored into the SKU text fields through the use of the a combinations (hereinafter "combo") selection process initiated

by clicking on the "Combos" button of the dialog box of FIG.

12. Once this button is clicked on, the dialog box of FIG. 13 is displayed. At this point the user can copy multiple

entries from combination tables by clicking a "Load" button

5 under each of three columns of text boxes. These combination tables may be previously generated by the same or a different user. Each combination table includes a plurality of columns each including a different combination of attribute values.

The text boxes may be filled as shown in the example of FIG.

10 13. If the user decides to clear the entries in the text boxes, he or she can click on one or more of the "Clear Style," "Clear Color" or "Clear Size" buttons or the user can directly create and/or edit the entries. The user can click on the "Make Combos" button to generate a combination table

15 listing all possible combinations of the various attribute values in the text boxes along with the parameters stored in temporary memory when the "Apply" button was selected. The "Save Combos" button can then be selected to save the

20 combination table. If desired, the "Advanced..." button can be clicked on to open the combination table in Excel® so that more powerful editing features can be employed. When the user

is finished with the combo process, he or she can select the "Done" button, whereupon the dialog box of FIG. 13 is closed.

It should be noted that the design of the dialog box of FIG. 13 will be different if values of different attributes are to be combined.

FIG. 14 illustrates the converting and cleaning process invoked by the block 168 of FIG. 9 and the block 198 of FIG. 11. A block 240 fetches the highlighted text from the page layout file. Next, a block 242 converts and cleans the text. In the illustrated embodiment, text cleaning software routines are written as a number of AppleScript<sup>®</sup> routines, each performing a different cleaning function. First, the block 242 can include a character counting function to truncate strings longer than a predetermined length. For example, if it is desirable to limit the product description records of the product table to only 30 characters to ensure proper display in a web page, then the block 242 could include a character counter which truncates any characters of the highlighted text of the page layout file that are beyond the thirtieth character. This character counting and truncation routine can be programmed using known routines.

Alternatively, the user could be prompted to reselect text from the page layout file.

Second, the block 242 may include an embedded control character elimination routine. In order to accomplish this cleaning function, an AppleScript<sup>®</sup> routine is written which stores the characterized text into a character string. The embedded character elimination routine iterates through each character within the string. As it iterates through each character, the routine determines if the ASCII value for the current string character is between a user-definable range of ASCII values. This range of ASCII values corresponds to the range of ASCII characters which will be properly displayed on-line. Characters outside of this desirable range, such as control characters, are removed or replaced with a more appropriate character (e.g., a tab may be replaced by a single space) by this second cleaning routine. As characters from the stored highlighted string are removed, the string is truncated by the removal. Alternatively, the embedded character elimination routine could compare the ASCII value of each character to a user-specified set of ASCII characters, instead of to a user-specified range of ASCII characters, and

remove those ASCII characters which match the user-specified ASCII characters.

Certain ASCII characters, like ®, ©, and ™ should not be removed, but rather replaced with codes recognizable by a web browser to display the character properly on-line. For example, the conversion program may contain routines which identify these ASCII characters and replace them with command strings recognizable in HTML, like &reg;, &copy; and &trade;, respectively. Other characters may be replaced through similar routines. In addition to these cleaning routines, any number of other cleaning routines, such as removing double spaces from highlighted text or replacing fraction strings with abbreviated characters (e.g., replacing "¼" with "1/2"). may also be implemented through similar AppleScript<sup>®</sup> programming routines.

As noted previously, the cleaning undertaken by the block 242 may vary depending upon which field or text box has been selected.

Before storing the cleaned highlighted text in a corresponding conversion field, the cleaned text is analyzed by a block 244 to determine whether it is valid. If the

cleaned text is valid, then it is placed into the appropriate field or text box. If the cleaned text does not satisfy the text criteria, an error message is displayed by a block 248. Control from the blocks 246 and 248 passes to the block 150 of FIG. 9 or the block 190 of FIG. 11.

While the process described above is an automated characterization and cleaning process, it should be noted that the user could manually enter data into the various fields or text boxes or could manually clean data stored in the conversion fields, as desired.

A detailed flow diagram of the process category selection block 158 of FIG. 9 is shown in FIG. 15. To input category information for a given object into the fields of the main dialog box of FIG. 2, the user clicks on one of the fields of the "Category Information" area, whereupon a block 280 of FIG. 15 displays a pop-up category dialog box (seen in FIG. 16). The category dialog box includes fields for indicating two different category assignments. A first set of fields (labeled "Main Category 1," "Sub Category 1" and "Detail Category 1") contains pull-down menus for entry of a first categorization of an object. A second set of fields (labeled



"Main Category 2," "Sub Category 2" and "Detail Category 2") includes pull-down menus that allow the user to specify a second categorization for the object. Selecting one of the main category pull-down menus causes a block 282 of FIG. 15 to display a list of all possible main categories as specified in the categories file. Once a main category has been chosen, a block 284 causes the associated sub-category pull-down menu to display only those sub-categories that are stored in memory under the selected main category. The user may then select a sub-category by clicking on same. Similarly, once a sub-category is selected, a block 286 causes the detail category field pull-down menu associated with the sub-category to display only those detail categories stored in memory under such sub-category. At this point the user may select the appropriate detail category by clicking on such detail category. An OK button and a Cancel button are provided for exiting the category dialog box. Upon exiting the category dialog box, the selected category information is populated into the "Category Information" fields of the main dialog box as shown in FIG. 2.

FIG. 17 illustrates the programming executed by the block 162 of FIG. 9. This process identifies which high-resolution images from the page layout file are to be converted to a web-ready format and displayed on-line. The process could also  
5 convert these identified images into small thumbnail images and/or cropped images and could further store "beauty shots," which are the enlarged images shown after a user has selected a thumbnail image. At a block 310, if the user has clicked on one of the high-resolution images in the displayed page  
10 layout, and subsequently clicks on the "Image Name" field, a dialog box is displayed identifying the file name and giving the user the opportunity to change the file name. Once the user has indicated acceptance of the original or modified file name, a block 312 stores the name of the image file as a  
15 string in the "Image Name" field of the main dialog box of FIG. 2. Typically, the stored image name string will be manually edited to remove any name extensions from the name. If, for example, the cucumber image of FIG. 1 were stored as a Photoshop® image with the name Cucumbers.tif, the .tif  
20 extension which indicates that the file is a tagged-image file format file is removed. This is done in part to allow other

name extensions, such as .gif and .jpg, to be appended onto the name upon image processing.

In addition, at any point during which the main dialog box of FIG. 2 is displayed and active, one or more image names identifying the image(s) derived from the high-resolution image can be stored in the fields under the "Image Names" heading of the "Detail Image(s)" area of the main dialog box of FIG. 2. Further, a description of each derived image can be inputted manually in the adjacent "Description" column of the main dialog box. A pull-down menu activated by clicking on the arrows of the button adjacent the label "Detail Image(s)" in the main dialog box can be used to select a "Counter" option to indicate the number of derived images stored for a given object. A button labeled "Edit Images" can be clicked on to allow editing of the description of the derived images in the "Description" field of the "Detail Image(s)" area of the main dialog box.

After all the appropriate fields of the main dialog box have been populated (e.g., as seen in FIG. 10), the operator can click on the "Save" button to save the data in the database file. The saving process is shown in FIG. 18. Every

conversion field is stored in a field of at least one database record, where all database fields for a given object are relationally linked with one another. This process begins at a block 336 which checks to determine whether all fields which must be filled with valid data are in fact so loaded. If this is not the case, an error message is generated and the saving process ends. On the other hand, if this is the case, then a block 340 checks to determine whether there are any remaining fields or text boxes of the main dialog box to process. If so, a block 342 fetches the contents of the next field or text box of the dialog box. A block 348 then places the field value into the database. Control then returns to the block 340.

Eventually, all of the field values are loaded into the database, whereupon control passes from the block 340 to a block 352, which checks to determine whether there are any images to process. If this is true, a block 354 places the name of the image into the database and a block 356 writes out the name translation information pertaining to the image into the name translation file. Control then returns to the block 352. Once all the images have been processed, a block 358

clears out all fields (except as noted below) in the main dialog box and resets all counters. Control then terminates. The pull-down values for the main category, sub category, and detail category of the category dialog box are not cleared because they may be used while converting page layout information for the next object.

As should be evident from the foregoing, the programming of FIG. 18 stores cleaned data and image names into the database file. A sample database file product table having fields stored therein is shown in FIGS. 19A, 19B, and 19C. The data stored therein corresponds to the data in the conversion fields of FIG. 10. Thus, the number B-54031 is stored in the first record of the field "item number." Other values are stored at least some of the remaining fields of the database. In this regard, it should be noted that the database contains data that will not be displayed on-line or which is not necessary to identify images to be displayed. For example, a field "tax exempt" is loaded with data, as are the categories fields "main," "sub" and "detail," even though such fields are not required to permit on-line display. These fields are provided to allow a user to track such information

for any desired purpose. In this regard, once the database is assembled it may be sent to the catalog publisher so that the publisher can track the information.

FIGS. 20A-20C illustrate the programming executed by the block 38 of FIG. 3. After all of the text information of the page layout file has been converted and cleaned, automated image processing functions are performed to convert specified high resolution images from the page layout file into images suitable for display in web browsers. The automated functions are undertaken by a program stored in a computer-readable memory, which may be the memory 26 of the computer system of FIG. 23. Alternatively, the instructions need not be stored in the same memory or run by the same processor that runs the conversion process discussed above. In fact, it may be desirable to have image processing performed on a different computer system than that illustrated in FIG. 23, although the computer system that undertakes the image processing would be similar or identical to that shown in FIG. 23.

Referring specifically to FIG. 20A, the same or a different user opens the image processing program, causing a block 380 to display an image dialog box illustrated in FIG.

21 and prompting the user to provide a specific location for the appropriate name translation file, (i.e., folder, directory or network location). Once this location is specified, a counter N is set equal to zero. (In the preferred  
5 embodiment the counter N is not used; however, employing the counter N in the description herein facilitates understanding. Also, some of the steps described in this and other portions of the specification do not find a one-to-one correspondence with the programming. See the Appendix to obtain a  
10 description of the programming that effects the steps undertaken by the present invention.) Thereafter, a block 382 opens the name translation file which, as noted above, provides a mapping from the names of the high-resolution image files that appear in the page layout file to the names of the  
15 derived images that will be displayed on-line. Alternatively, as shown in FIG. 20, the image processing program could execute a set of instructions for automatically creating a first list of high-resolution image files stored in a particular location, and which further develops a list of  
20 thumbnail and cropped image names from the first list.

In either case, a block 384 opens a "cropped images file" containing a listing of the names of previously cropped images. If a cropped images file is not found (because it has not been yet created), an empty cropped images file is created at this point. This cropped images file may be stored at the location of the high-resolution files, or may be stored at another directory, folder or network location. A block 386 then compares the name N of the name translation file to the names stored in the cropped images file. If the name N of the name translation file is not in the cropped images file, then it has been determined that the image identified by the name N of the name translation file must be cropped and/or otherwise processed. Accordingly, a block 390 opens the high resolution image in a native image processing application, such as Adobe Photoshop®. As seen in FIG. 21, the image dialog box has five buttons: "Next," "Skip," "Start Batch," "Thumbnail" and "Quit." Because high-resolution images often cannot be loaded by a browser in a sufficiently short time and because of display resolution limitations, it may be desirable to crop the high resolution images and/or reduce the size in pixels of the images so that only a portion thereof will be displayed.



At a block 392 (FIG. 20B) desired non-automated modifications to the high resolution image are undertaken by the user using the native image processing software.

After the desired non-automated modifications are  
5 complete, control pauses at a block 394 to allow the operator to select any one of the "Next," "Skip," "Start Batch," "Thumbnail" or "Quit" buttons. If the "Next" button is clicked on, then a block 398 executes specified changes to the image that can be programmed to be undertaken by the image  
10 processing application (i.e., the block executes "scriptable", or otherwise programmable changes to the image.) In the preferred embodiment, the block 398 converts the image file from a CMYK format (consisting of values for the printing primary colors of cyan, magenta, yellow and key (or black)) to  
15 a three-color RGB (e.g, red, green and blue) format suitable for computer monitor display. If desired, an indexed color format file could also be derived from the RGB format file. After programmed changes have been effected, a block 400 saves the modified image(s) as intermediate image(s) and names such  
20 files with a mapped name, which is mapped from the name of the high resolution image file. In the illustrated embodiment,

the intermediates file(s), however, have no extension. The high resolution image file is then closed and a block 402 appends the name N to the cropped images file, indicating that the processing of this image has been completed. Control then  
5 returns to a block 430 of FIG. 20A.

If the block 396 determines that the "Next" button has not been selected, a block 404 checks to determine whether the "Thumbnail" button has been clicked on. If this is the case, a block 406 executes scriptable changes to the image (like the  
10 block 398) and a block 408 converts the image to thumbnail size. The high-resolution image is changed to thumbnail size through the use of image processing algorithms in the image processing software, which can (for example) reduce an 800x600 pixel image to a 40x30 pixel image. A block 410 then  
15 automatically performs an unsharp masking routine using the native application and a block 412 saves the resulting thumbnail image file(s) with mapped filename(s) mapped from the name of the high resolution image, but with the extension ".gif." Thereafter, a block 414 closes the thumbnail image  
20 file, a block 416 reopens the original image file and a block 418 appends the name N in the name translation file to an

additional file called the "Done Images File," which lists all images that have been converted into a thumbnail. Control then returns to the block 392 of FIG. 20B to permit the corresponding high-resolution image to be processed in Photoshop® for custom modifications. This allows a set of custom modifications, separate from the custom modifications made in creating the thumbnail image, to be performed in creating the cropped image.

If desired, if an image name appears with a detail image extension, (i.e., if a file has a name with one of the extensions ".dn," where n=1, 2, 3...) then the programming could provide a dialog box indicating to the operator that the file is a detail image and, therefore, that no thumbnail image will be created by the programming of Fig. 22.

The above customizations of both the thumbnail image and the cropped image can be suspended by selecting the Quit button from the image dialog box. Specifically, if the block 404 of FIG. 20B determines that the "Thumbnail" button has not been selected, a block 420, FIG. 20C, checks to determine whether the "Skip" button has been clicked on. If this is the case, control returns to the block 430 of FIG. 20A. The block

430 increments the value of N and control passes to the blocks  
386 and 388. At some point in the process all of the image  
files will have been processed and a block 428 passes control  
to a block 432, at which the automated processing routine of  
5 FIG. 22 is invoked. Following the block 432, a block 434  
determines whether the user has selected an option "Open  
Folder" in the "Files" heading of a menu bar that is displayed  
by the operating system (Apple OS). If this has been  
selected, control passes to the block 380. Otherwise, a block  
10 436 checks to determine whether the user has selected an  
option "Quit" in the "Files" heading of the menu bar. If this  
is true, program execution terminates. If this is not true,  
control returns to the block 434.

Referring again to FIG. 20C, if the block 420 determines  
15 that the "Skip" button has not been selected, a block 422  
checks to determine whether the "Start Batch" button has been  
selected. If this button has been clicked on, then the user  
has requested that automated image processing be undertaken  
and hence the automated processing routine shown in FIG. 22 is  
20 invoked by a block 424. After the automated processing

routine is complete, control passes to the block 434 of FIG. 20A.

If the block 422 determines that the "Start Batch" button has not been selected, a block 426 determines whether the "Quit" button has been clicked on. If this is the case, further program execution is terminated. Otherwise, control returns to the block 394 of FIG. 20B.

Referring now to FIG. 22 the automated procedures can be facilitated by an automation tool, such as PreFab Player<sup>®</sup> from PreFab Software, Inc. of Westford, MA, which allows the execution of pre-defined actions such as button selections, keystrokes, and control commands. Preferably, the automated procedures at least open each RGB file from the cropped files list, apply an unsharp masking to the image, apply an image compression such as a JPEG compression to the image, and save the compressed file in the compressed image format. If the file is not already in the "Done Files" list, the same RGB file is then re-opened, scaled to thumbnail size, sharpened, converted into an indexed color file and saved as a compressed image file, such as a GIF file, after which the file name is stored in the "Done Files" list. Other automated actions may be carried out in a similar manner. The thumbnail and cropped

images created in this manner and stored in the proper location correspond to the image names stored by the programming of FIG. 18. Furthermore, the images are sized and image processed for proper display in the on-line catalog.

5           Specifically, as seen in FIG. 22, a block 450 reads the next name from the cropped images file (if this is the first pass through the programming of FIG. 22, the block 450 reads the first name in the file.) A block 452 then checks to determine whether all images in the cropped images file have  
10           been processed. If so, then execution of the programming of FIG. 22 ends. If not, a block 454 performs unsharp masking on the file contents and a block 456 performs a JPEG compression and saves the resulting file with the extension ".jpg." A block 458 then determines whether the file is a detail image.  
15           If this is not the case, a block 462 checks to determine whether the file name is in the Done Files list. If this is also not the case, a series of blocks 462, 464, 466 and 468 changes the size of the image to thumbnail, performs unsharp masking, executes scriptable changes (identical or similar to  
20           the block 398 of FIG. 20b, if these changes have not already been made to the image) and saves the resulting file(s) with

the extension ".gif." A block 470 then closes the image file(s).

Following the block 470, or following the blocks 458 and 460 if the file is a detail file or if the file is in the Done Files list, a block 472 deletes the intermediate file(s) (i.e., the file(s) that were stored with no extension). A block 474 then determines whether the Done Files list includes the file name. If this is not the case, a block 476 adds the file name to the Done Files list and control returns to the block 450. The block 476 is skipped and control returns directly to the block 450 if the block 474 determines that the Done Files list does not include the file name.

It should be evident from the foregoing that the present invention converts the unstructured and untagged data in the page layout file into searchable data related to an object.

Numerous modifications to the present invention will be apparent to those skilled in the art in view of the foregoing description. Accordingly, this description is to be construed as illustrative only and is presented for the purpose of enabling those skilled in the art to make and use the invention and to teach the best mode of carrying out same.

The exclusive rights of all modifications which come within the scope of the appended claims are reserved.



What we claim is:

1. A system for converting a page layout file into a database for use by a display program which converts the database for display in a medium, comprising:

5 means for opening the page layout file;  
means responsive to selection of a portion of the page layout file for converting the portion into data compatible with the medium; and

means responsive to the converting means for  
10 assembling the data into the database.

2. The system of claim 1, wherein the converting means includes means for removing codes from the portion that are incompatible with the medium.

15 3. The system of claim 1, wherein the converting means includes means for adding characters to the data, wherein the characters are compatible with the medium.

4. The system of claim 1, wherein the page layout file includes image and text portions and wherein the converting means includes means responsive to selection of the image portion for developing a display-ready image file from the image portion and means responsive to selection of the text portion for deriving a text file from the text portion wherein the text file is compatible with the medium.

5. The system of claim 4, wherein the developing means comprises means for producing derived images from the image portion.

6. The system of claim 5, wherein the producing means includes means for invoking an image processing program to allow editing of the image portion, means for generating an indication that editing of the image portion is complete and means responsive to the generating means for retrieving a further image for editing.

7. The system of claim 6, further including means responsive to the generating means for performing automated

image processing on the image portion before the further image portion is retrieved by the retrieving means.

8. The system of claim 1, wherein the page layout file  
5 comprises a representation of a catalog page having an object therein, wherein the object has attributes specified in the page layout file and further including means for automatically generating combinations of attributes.

10 9. The system of claim 1, wherein the page layout file comprises a representation of a catalog page having an object therein and the portion includes unstructured and untagged data, and wherein the converting means converts the  
15 unstructured and untagged data into searchable data related to the object.

10. A software system for converting a page layout file into a database for use by a display program which converts the database for display in a certain medium, comprising:

a computer-readable medium; and

5 a software program stored in the computer-readable medium and including

a first routine that opens the page layout file,

10 a second routine responsive to selection of a portion of the page layout file and that converts the portion into data compatible with the certain medium, and

15 a third routine responsive to the second routine and that assembles the data into the database.

11. The software system of claim 10, wherein the second routine includes a routine that removes codes from the portion that are incompatible with the certain medium.

12. The software system of claim 10, wherein the second routine includes a routine for adding characters to the data, wherein the characters are compatible with the certain medium.

13. The software system of claim 10, wherein the page layout file includes image and text portions and wherein the second routine includes a fourth routine responsive to selection of the image portion and that develops a display-ready image file from the image portion and a fifth routine responsive to selection of the text portion and that derives a text file from the text portion wherein the text file is compatible with the certain medium.

14. The software system of claim 13, wherein the fourth routine includes a sixth routine that produces derived images from the image portion.

15. The software system of claim 14, wherein the sixth routine includes a seventh routine that invokes an image processing program to allow editing of the image portion, an eighth routine that generates an indication that editing of

the image portion is complete and a ninth routine responsive to the generating means and that retrieves a further image for editing.

16. The software system of claim 15, further including a tenth routine responsive to the eighth routine and that performs automated image processing on the image portion before the further image portion is retrieved by the retrieving means.

17. The software system of claim 10, wherein the page layout file comprises a representation of a catalog page having an object therein, wherein the object has attributes specified in the page layout file and further including a combinations routine that automatically generates combinations of attributes.

18. The software system of claim 10, wherein the page layout file comprises a representation of a catalog page having an object therein and the portion includes unstructured and untagged data, and wherein the second routine converts the unstructured and untagged data into searchable data related to the object.

19. A software system for converting a page layout file representing a catalog page into a database for use by a program which converts the database into web pages encoded in an Internet web format, the catalog page including a text portion and an image portion wherein the text portion and the image portion relate to a catalog object, comprising:

a computer-readable medium; and

a software program stored in the computer-readable medium and including

a first routine that opens the page layout file,

a second routine responsive to selection of one of the text portion and the image portion and that converts the portion into data compatible with the web format, and

a third routine responsive to the second routine and that assembles the data into the database and relates such data to the object.



20. The software system of claim 19, wherein the second routine includes a routine that removes codes from the text portion that are incompatible with the web format.

21. The software system of claim 19, wherein the second routine includes a routine for adding characters to the data, wherein the characters are compatible to the web format.

22. The software system of claim 19, wherein the second routine includes a fourth routine responsive to selection of the image portion and that develops a display-ready image file from the image portion and a fifth routine responsive to selection of the text portion and that derives a text file from the text portion wherein the text file is compatible with the web format.

23. The software system of claim 22, wherein the fourth routine includes a sixth routine that produces derived images from the image portion.

24. The software system of claim 23, wherein the sixth routine includes a seventh routine that invokes an image processing program to allow editing of the image portion, an eighth routine that generates an indication that editing of the image portion is complete and a ninth routine responsive to the generating means and that retrieves a further image for editing.

25. The software system of claim 24, further including a tenth routine responsive to the eighth routine and that performs automated image processing on the image portion before the further image portion is retrieved by the retrieving means.

26. The software system of claim 19, wherein the object has attributes specified in the page layout file and further including a combinations routine that automatically generates combinations of attributes.

27. The software system of claim 19, wherein the text portion includes unstructured and untagged data, and wherein

the second routine converts the unstructured and untagged data into searchable data related to the object.

28. A system for converting a page layout file for display in a medium, comprising:

means for opening the page layout file;

means responsive to selection of a portion of the page layout file for converting the portion into data compatible with the medium;

means responsive to the converting means for assembling the data into a database; and

a display program which converts the database into page description files for display in the medium.

29. The system of claim 28, wherein the converting means includes means for removing codes from the portion that are incompatible with the medium.

30. The system of claim 29, wherein the converting means includes means for adding characters to the data, wherein the characters are compatible with the medium.

31. The system of claim 28, wherein the page layout file includes image and text portions and wherein the converting means includes means responsive to selection of the image portion for developing a display-ready image file from the image portion and means responsive to selection of the text portion for deriving a text file from the text portion wherein the text file is compatible with the medium.

32. The system of claim 31, wherein the developing means comprises means for producing derived images from the image portion.

33. The system of claim 32, wherein the producing means includes means for invoking an image processing program to allow editing of the image portion, means for generating an indication that editing of the image portion is complete and means responsive to the generating means for retrieving a further image for editing.

34. The system of claim 33, further including means responsive to the generating means for performing automated

image processing on the image portion before the further image portion is retrieved by the retrieving means.

35. The system of claim 28, wherein the page layout file comprises a representation of a catalog page having an object therein, wherein the object has attributes specified in the page layout file and further including means for automatically generating combinations of attributes.

36. The system of claim 28, wherein the page layout file comprises a representation of a catalog page having an object therein and the portion includes unstructured and untagged data, and wherein the converting means converts the unstructured and untagged data into searchable data related to the object.

37. The system of claim 28, wherein the medium comprises the Internet and wherein the page layout file represents a catalog page and wherein the display program develops at least one web page.

38. A method of converting a page layout file for display in a medium, the method comprising the steps of:

opening the page layout file;

in response to selection of a portion of the page layout file, converting the portion into data compatible with the medium;

assembling the data into a database; and

using a display program to convert the database into page description files for display in the medium.

39. The method of claim 38, wherein the step of converting includes the step of removing codes from the portion that are incompatible with the medium.

40. The method of claim 39, wherein the step of converting further includes the step of adding characters to the data, wherein the characters are compatible with the medium.

41. The method of claim 38, wherein the page layout file includes image and text portions and wherein the step of

converting includes the steps of developing a display-ready image file from the image portion and deriving a text file from the text portion wherein the text file is compatible with the medium.

42. The method of claim 41, wherein the step of developing includes the step of producing derived images from the image portion.

43. The method of claim 42, wherein the step of producing includes the steps of invoking an image processing program to allow editing of the image portion, generating an indication that editing of the image portion is complete and retrieving a further image for editing.

44. The method of claim 43, further including the step of performing automated image processing on the image portion before the further image portion is retrieved by the retrieving means.




45. The method of claim 38, wherein the page layout file comprises a representation of a catalog page having an object therein, wherein the object has attributes specified in the page layout file and further including the step of automatically generating combinations of attributes.

46. The system of claim 38, wherein the page layout file comprises a representation of a catalog page having an object therein and the portion includes unstructured and untagged data, and wherein the step of converting converts the unstructured and untagged data into searchable data related to the object.

1 / 26

12/13.qxd



**Phlox Watercolor Memories**—Heat tolerant

B-15998 8 Pitt  
(35 seeds) \$1.95  
Two \$3.45

**Watercolor Memories** ◊

Phlox drummondii. These fragrant beauties are covered with flowers from June until frost. Our mix has subtle new shades never seen in phlox: sandy peach and buff yellow tones mixed with sunset pinks and soft lavender blues. They're very easy to grow too. Start early indoors, or sow the seed where you want plants to

**Tomato SuperTasty**—Tastiest in our his ◊

**Fun 'n Sun** ◊

Hybrid Sunflower Mix. Our designer blend of 10 varieties will show you why everybody's so excited about sunflowers! You get singles and fluffy doubles. Some blooms are a hefty 8", some only 3". There are wine reds, rich golds, soft yellows, white chiffon. Bicolors, too. They'll last 7 days in vase. All are pollen-free and won't shed on tabletops. Plants are only 4-6' tall. Abundant blooms for well over two months!

**SUNSEE EXCLUSIVE**  
B-57026 8 Pitt (30 seeds) \$2.95 Two \$4.95

**Sunflower**  
Fun 'n Sun ◊

**Cucumber Sugar Crunch**—◊

bloom—near paths or in containers. Flowering begins in six weeks. 18" tall. **SUNSEE EXCLUSIVE**  
B-36134 6 Pitt (50 seeds) \$2.95 Two \$4.95  
**VALUE!** B-29080 58 Watercolor Memories plants \$29.95

**Sugar Crunch** ◊

Hybrid Cucumber. 57 days. This little cucumber does for cucles what the Sugar Snap did for peas. They're best when just 4" or 5" long. The remarkably sweet flavor and crisp, crunchy texture means you can fix cucumbers in all sorts of new ways. It has the smooth tender skin of a greenhouse cucumber, yet the petite size of a pickler. The plant makes mostly female flowers, so expect an incredible 80 to 20 cucles per plant. Vines are compact (though not a bush) and resistant to mosaic virus, and powdery and downy mildews. **SUNSEE EXCLUSIVE**  
B-54031 6 Pitt (30 seeds) \$2.95 Two \$4.95

**SuperTasty** ◊

Hybrid Tomato. 70 days. Rich, tangy flavor, extremely juicy, solid texture, and saturated with a deep red color inside and out. Bushy, disease free, vigorous and productive. Space saving habit, with minimal staking required. 7-9 oz. fruits. **SUNSEE EXCLUSIVE**  
B-57026 8 Pitt (30 seeds) \$2.95 Two \$4.95  
B-36732 Pitt & SEED 'n START Kit ◊

**Bean French Fries**—Delicious gourmet beans

FIG. 1

2 / 26

**QuarkXPress Converter**

☒ Item Number  
☐ Item Name  
☐ Headline  
☐ In Description  
☐ Image Name  
☐ Keywords  
☐ Description

Detail Image(s): None #Image(s): 1 Edit Images

	Image Names	Description
1		
2		

**Category Information:**

	Main	Sub	Detail
#1			
#2			

**SKU Information:**

	SKU	Style	Price	Color	Size	Sale P	Quan	Quan N	TP	W	M
1											
2											
3											
4											
5											
6											

Reset Excel Edit on SKU WS Reset Save  
Save Excel Save SKU WS Done

FIG. 2

3 / 26

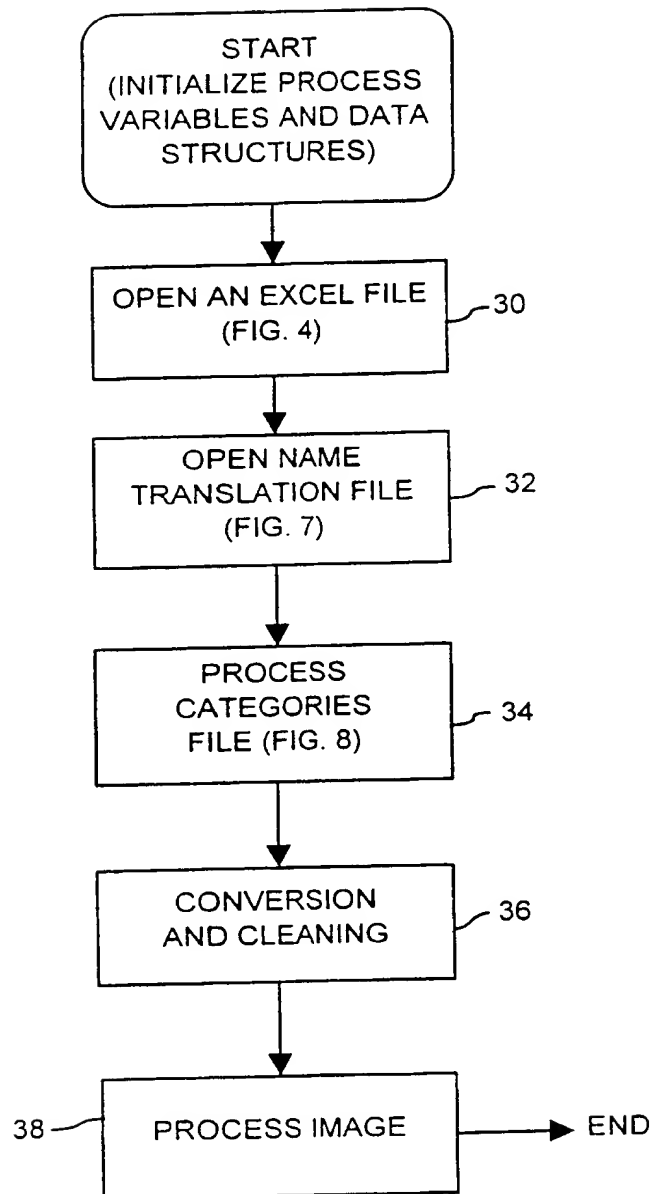


FIG. 3

4 / 26

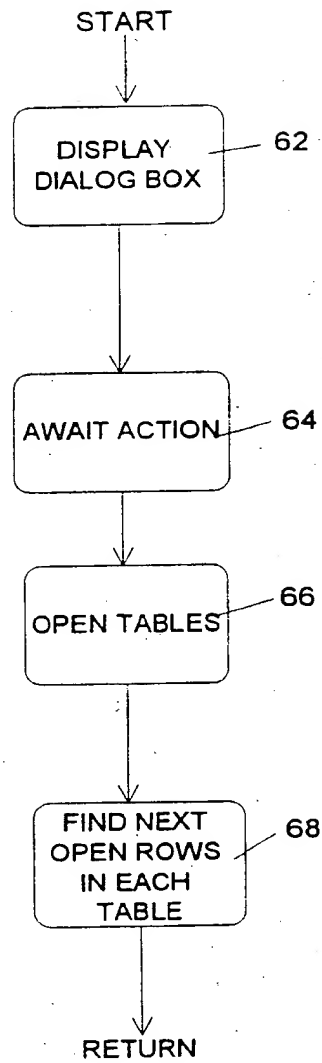


FIG. 4

PRODUCT TABLE 1

Item number	Item Name	Keywords	Catalog ID	Manufacturer Name	Manufacturer Logo	Main Cat Code 1	Sub Cat Code 1
-------------	-----------	----------	------------	-------------------	-------------------	-----------------	----------------

FIG. 5A

PRODUCT TABLE 2

Detail Cat Code 1	Main Cat Code 2	Sub Cat Code 2	Detail Cat Code 2	Headline	Thumbnail Description	Thumbnail Image
-------------------	-----------------	----------------	-------------------	----------	-----------------------	-----------------

FIG. 5B

PRODUCT TABLE 3

Product description	Product Page Image	Publish	Default Price	Tax Exempt
---------------------	--------------------	---------	---------------	------------

FIG. 5C

6/26

SKU TABLE 1

Promo Name	Promo Desc	Promo Rank	Active	Date Start	Date End	Sale	Gift Wrap Price	Monogramming Price	Personalization Price	Date Effective
------------	------------	------------	--------	------------	----------	------	-----------------	--------------------	-----------------------	----------------

FIG. 6



7 / 26

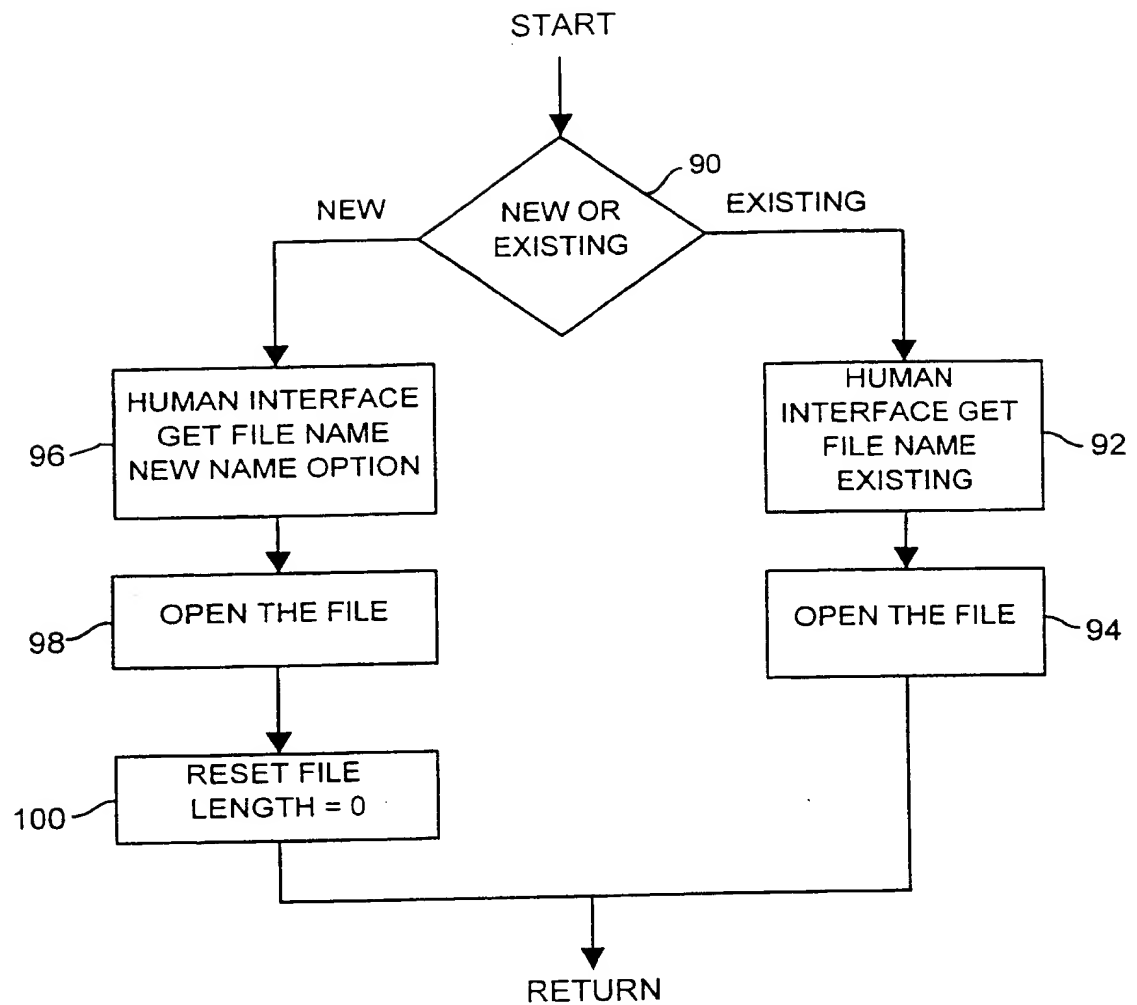


FIG. 7

8/26

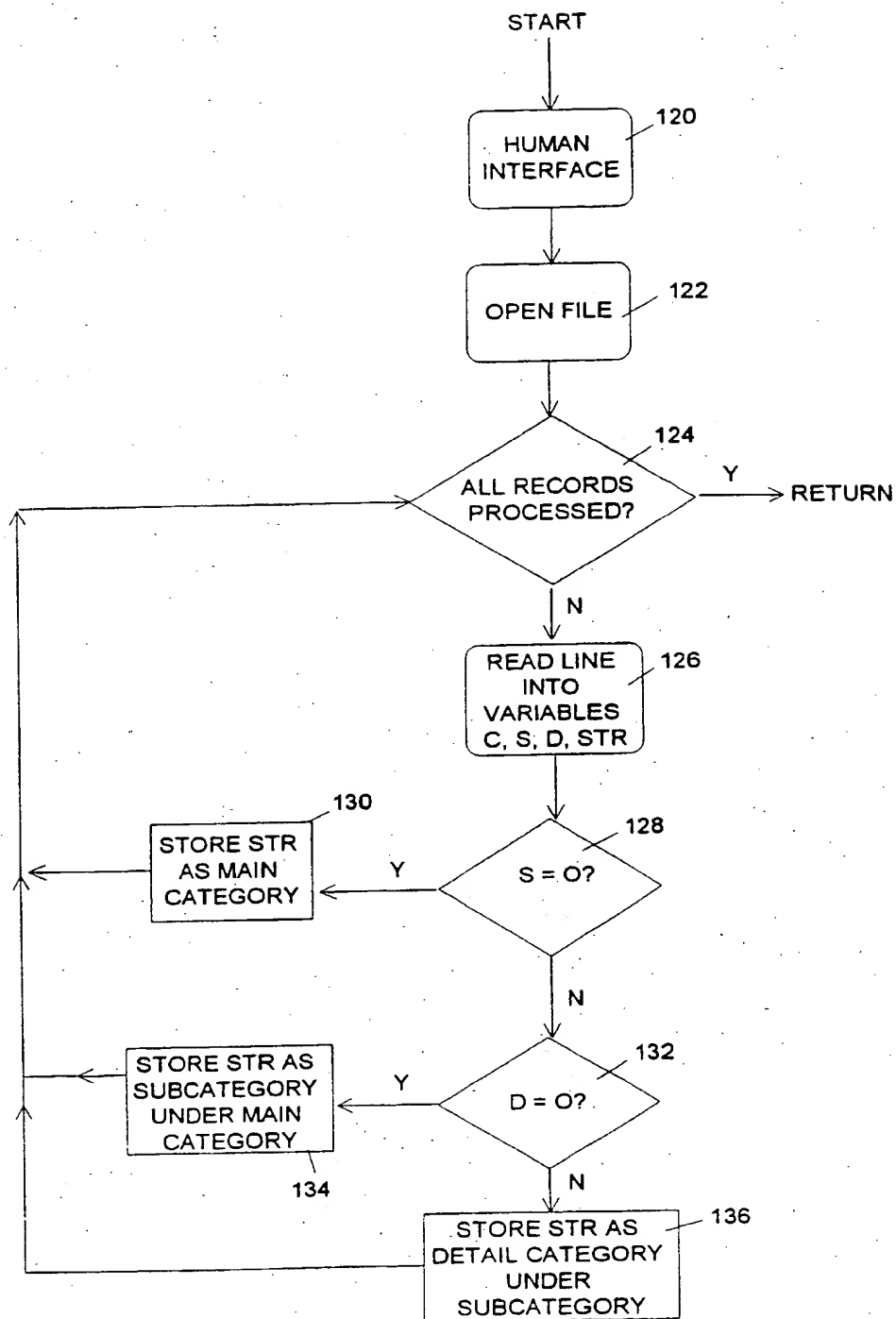


FIG. 8

9 / 26

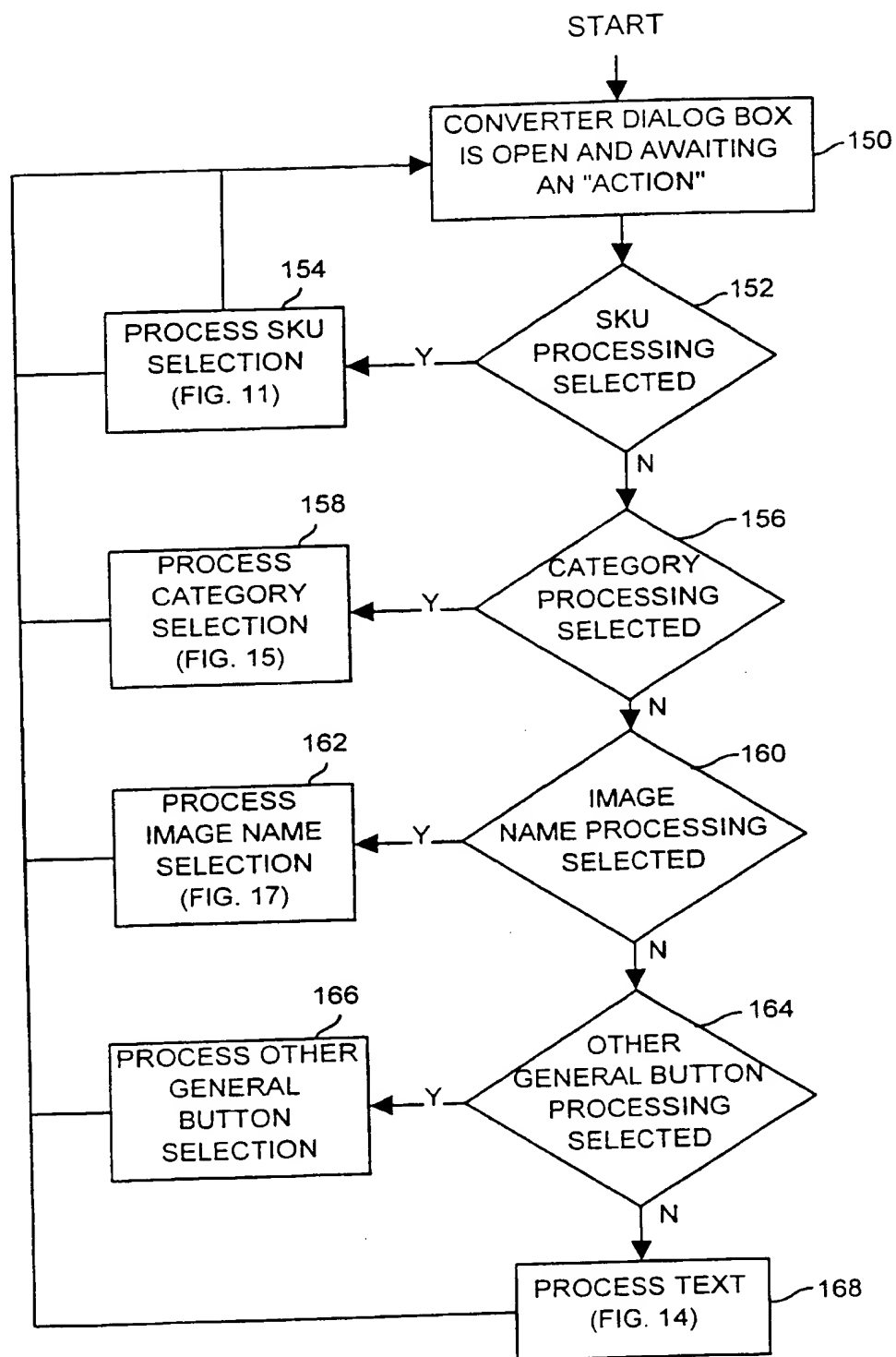


FIG. 9

10 / 26

**QuarkHPress Converter**

☐ Item Number: B-54031  
☐ Item Name: Sugar Crunch  
☐ Headline: Hybrid Cucumber  
☐ In Description: Cucumber  
☒ Image Name: A736194 Burpee Gardens 97'  
☐ Keywords: Cucumber, Garden, Vegetable, Summer  
☐ Description: This little cucumber does for cukes what the Sugar Snap did for peas. They're best when just 4 to 5 inches long. The remarkably sweet flavor and crisp, crunchy texture means you can fix cucumbers in all

Detail Image(s): None #Image(s): 1 Edit Images

	Image Names	Description
1	A736194 Burpee Gardens 97' copy	
2	A736194 Burpee Gardens 97' copy	

**Category Information:**

	Main	Sub	Detail
#1	Lawn & Garden	Vegetables	
#2	Lawn & Garden	Burpee New & Favorites	

**SKU Information:**

	SKU	Style	Price	Color	Size	Sale P	Quan	Quan N	TP	W	M
1											
2											
3											
4											
5											
6											

Reset Excel Edit on SKU WS Reset Save  
Save Excel Save SKU WS Done

FIG. 10

11 / 26

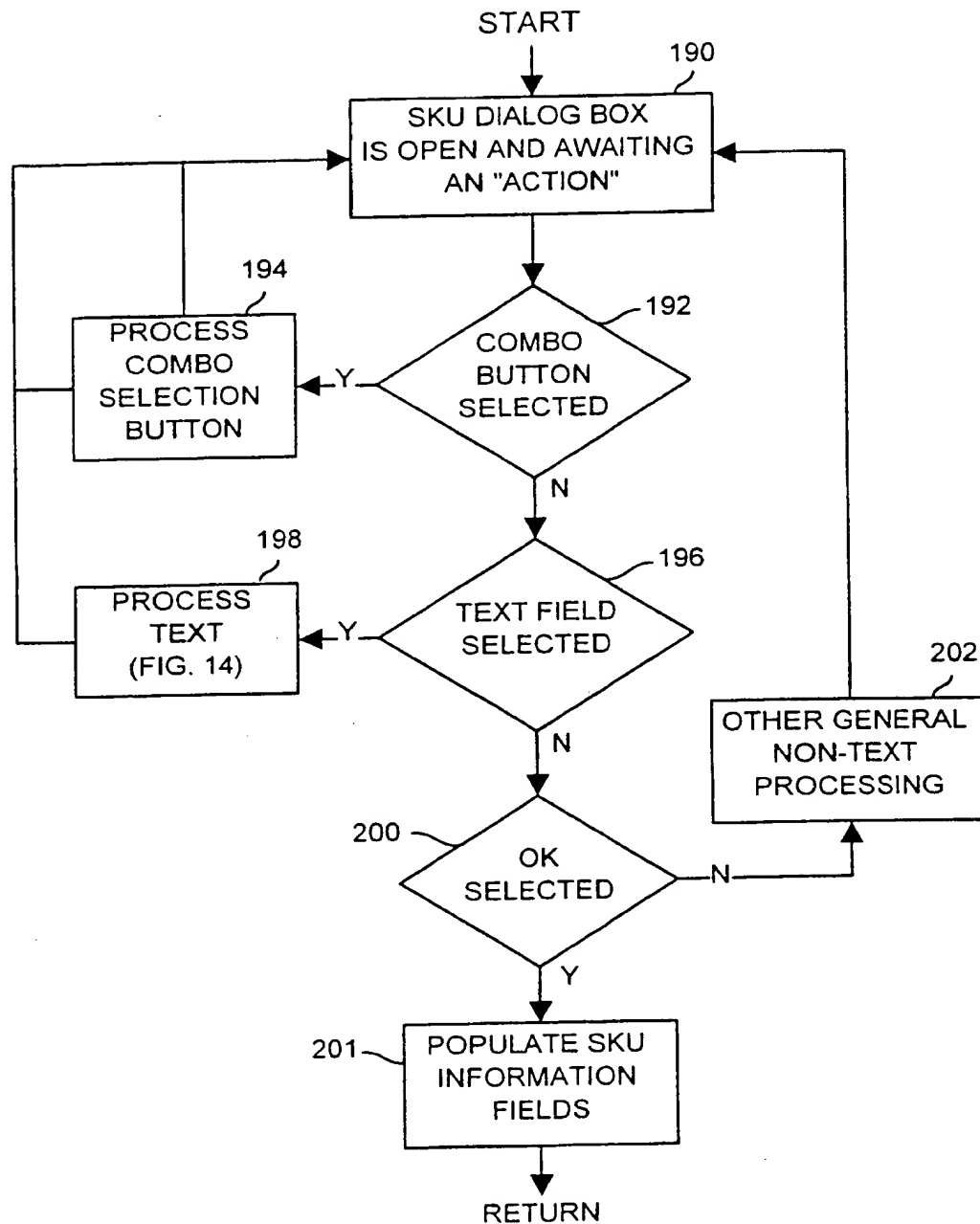


FIG. 11

12 / 26

SKU Information	
Item Number:	B-54031
<input type="radio"/> SKU Number:	<input type="text"/>
<input type="radio"/> Style:	<input type="text"/>
<input type="radio"/> Price:	<input type="text"/>
<input type="radio"/> Color:	<input type="text"/>
<input type="radio"/> Size:	<input type="text"/>
<input type="radio"/> Sale Price:	<input type="text"/>
<input type="radio"/> Quantity Price:	<input type="text"/>
<input type="radio"/> Quantity Needed:	<input type="text"/>
<input type="radio"/> Weight:	<input type="text"/>
<input type="checkbox"/> Thumbnail Price	<input type="button" value="Apply"/>
Copies: <input type="text" value="1"/>	
<input type="button" value="Delete"/>	<input type="checkbox"/> Mark Item
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>
<input type="button" value="Combo"/>	

FIG. 12



14 / 26

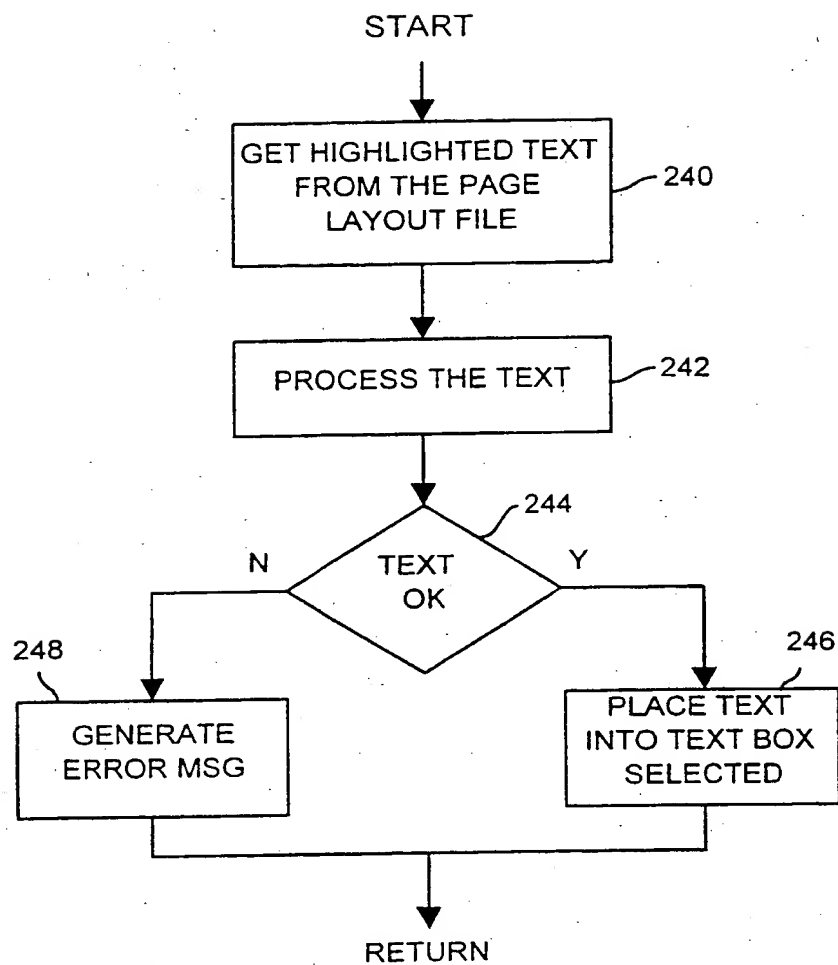


FIG. 14



15 / 26

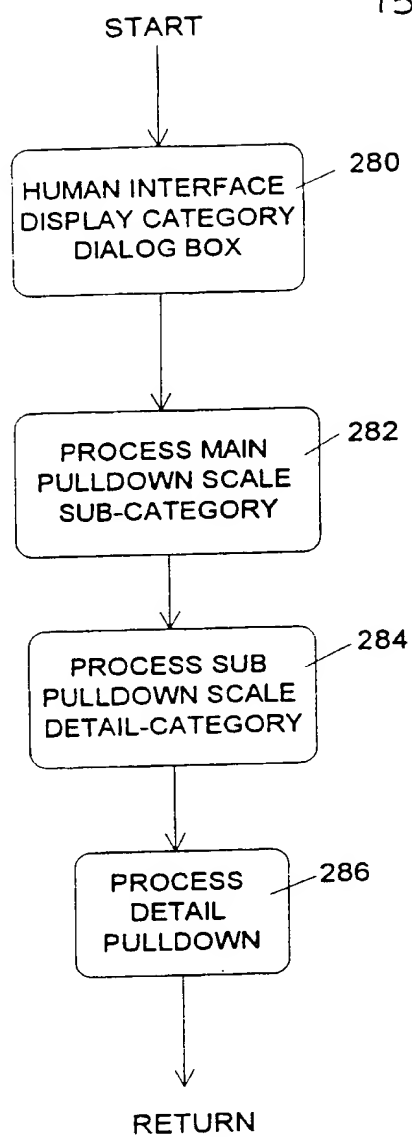


FIG. 15

16 / 26

Main Category 1:	Lawn & Garden
Sub Category 1:	Burpee New & Favorites
Detail Category 1:	Other Flowers
Main Category 2:	None
Sub Category 2:	None
Detail Category 2:	None

OK Cancel

FIG. 16

17 / 26

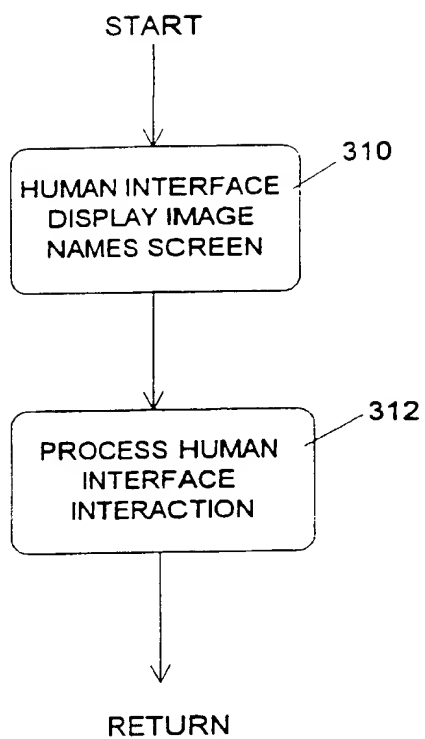


FIG. 17

18 / 26

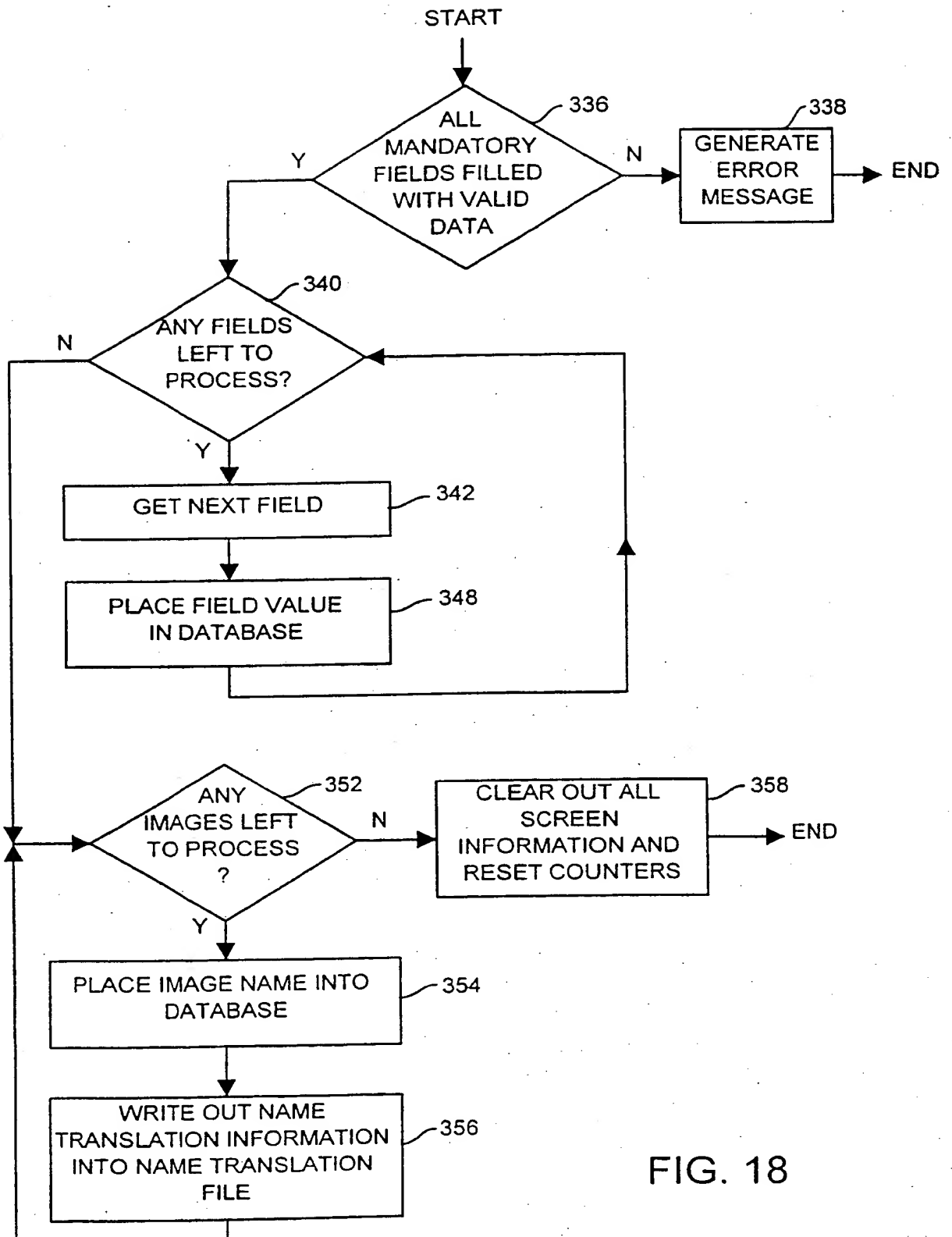


FIG. 18

Item number	Item Name	Keywords	Catalog ID	Manufacturer Name	Manufacturer Logo	Main Cat Code 1	Sub Cat Code 1
B-54031	Sugar Crunch	Cucumber, Garden, Vegetable, Summer				Lawn & Garden	Vegetables

FIG. 19A

Detail Cat Code 1	Main Cat Code 2	Sub Cat Code 2	Detail Cat Code 2	Headline	Thumbnail Description	Thumbnail Image
Cucumbers	Lawn & Garden	Burpee New & Fav	Cucumbers	Hybrid Cucumber	Cucumber	A736194 Burpee Gar

FIG. 19B

Product description	Product Page Image	Publish	Default Price	Tax Exempt
This little cucumber does	A736194 Burpee Gardens	1		0

FIG. 19C

20 / 26

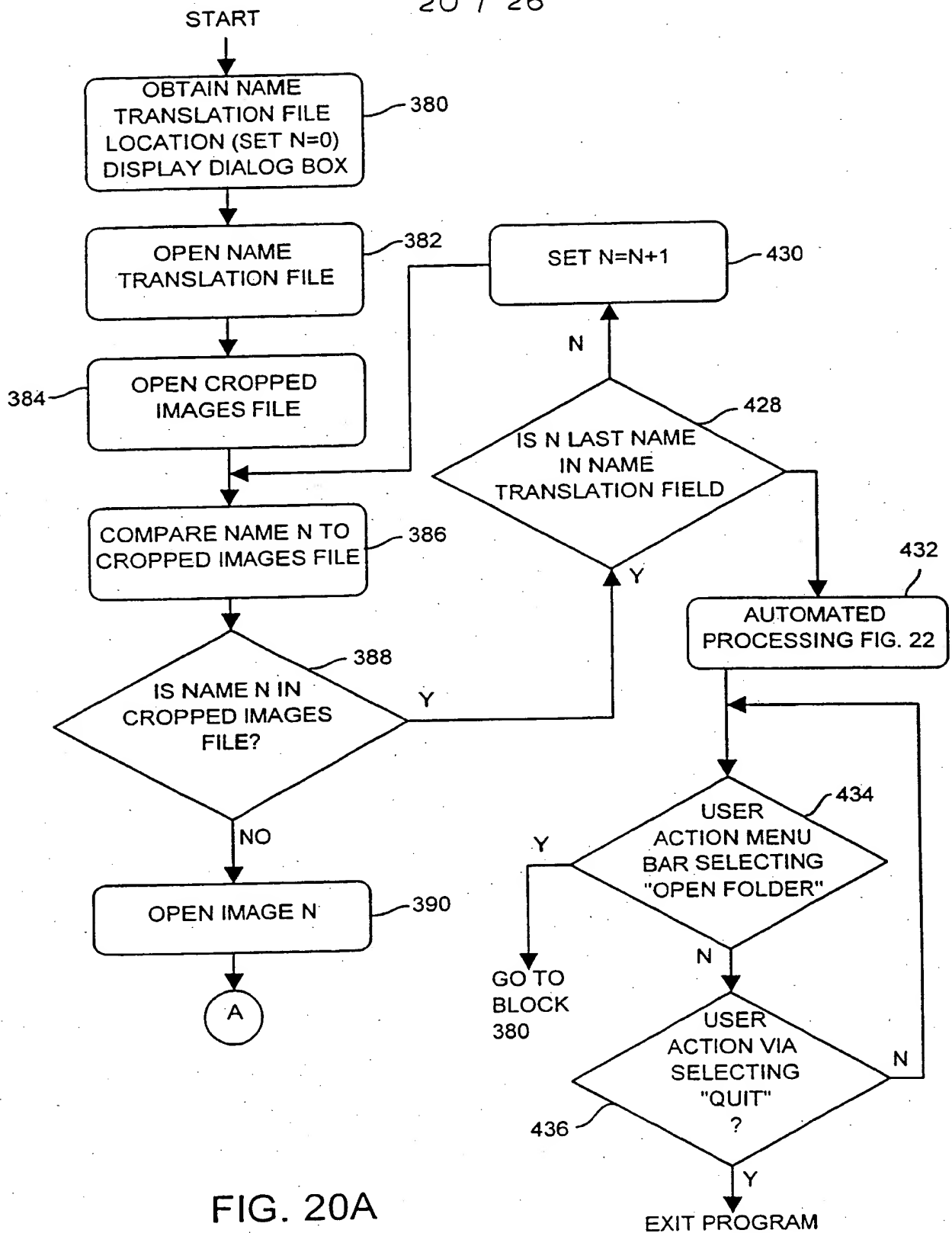


FIG. 20A

21 / 26

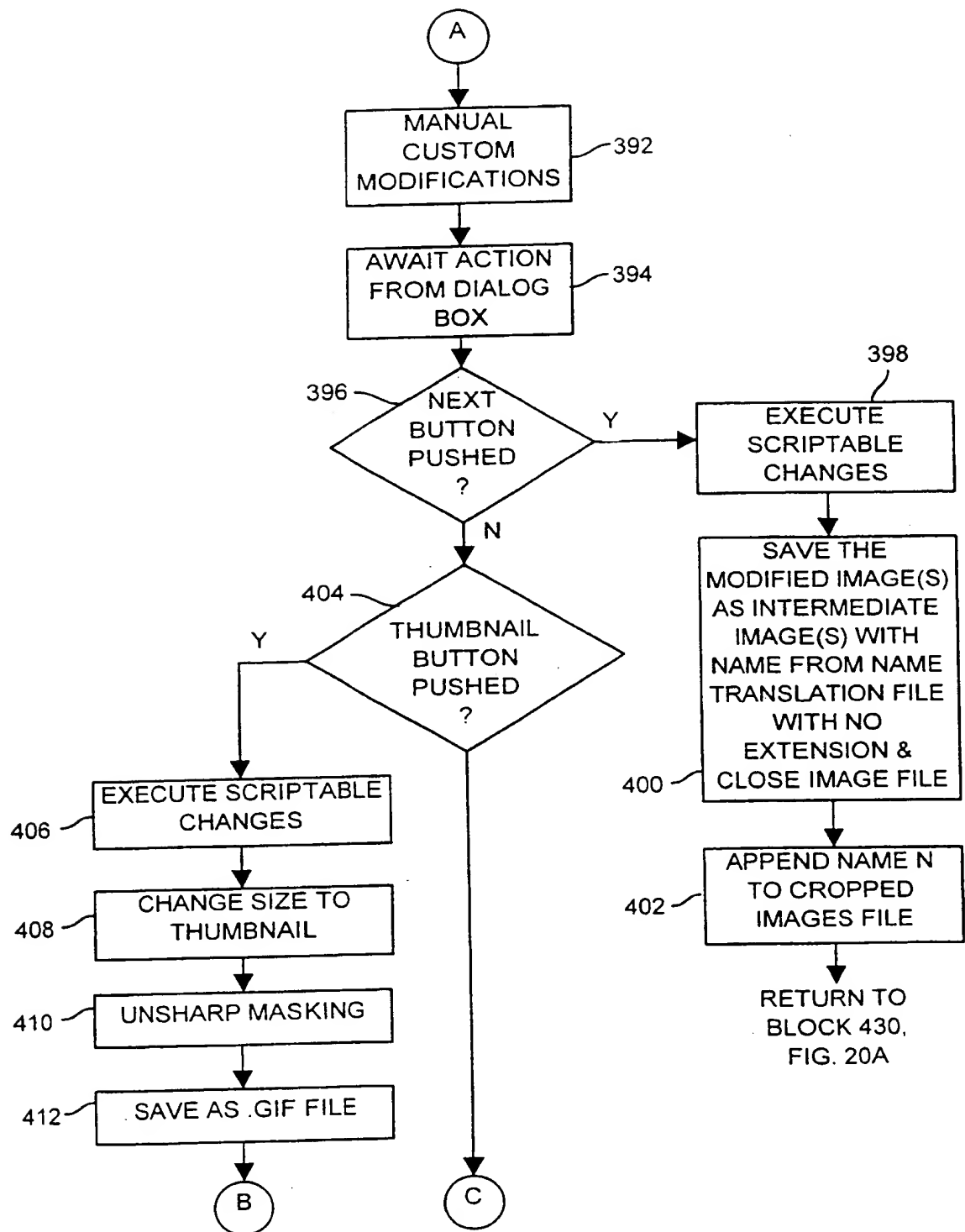


FIG. 20B

22/26

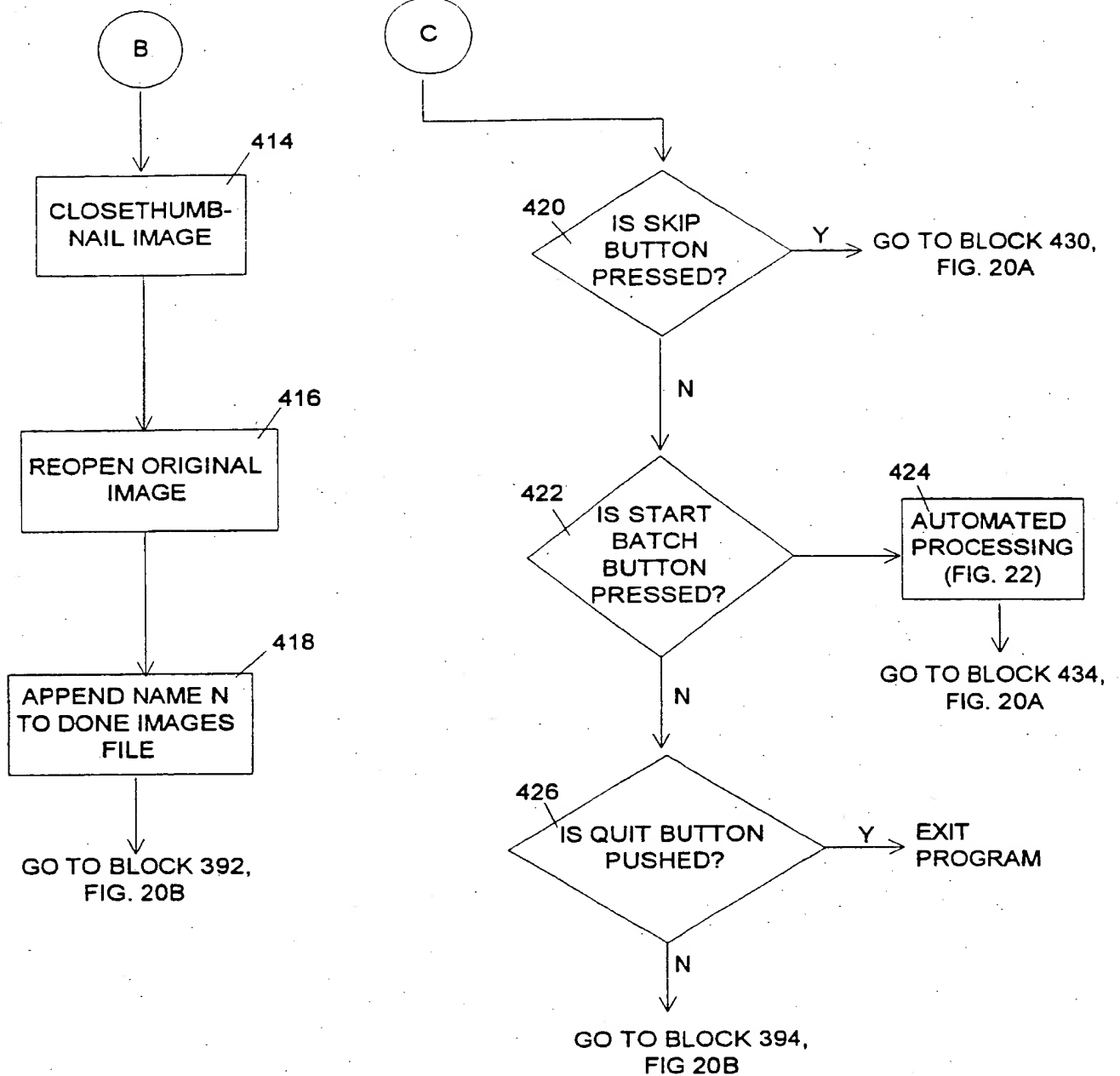


FIG. 20C



23 / 26

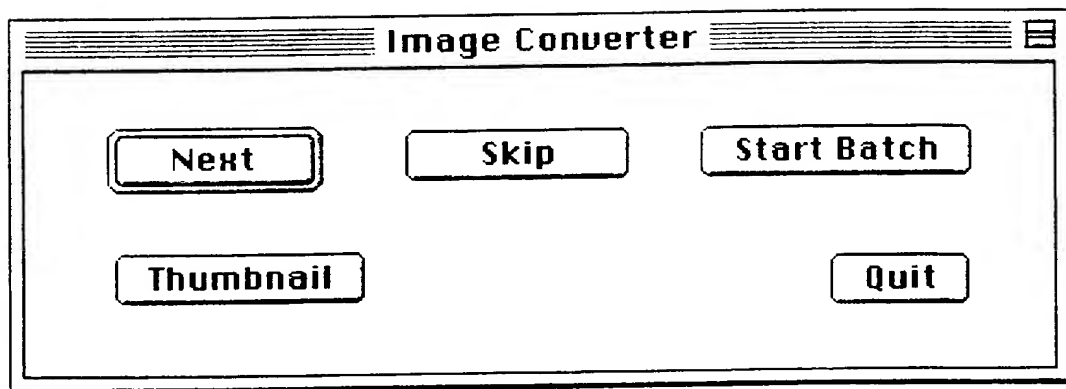


FIG. 21

24 / 26

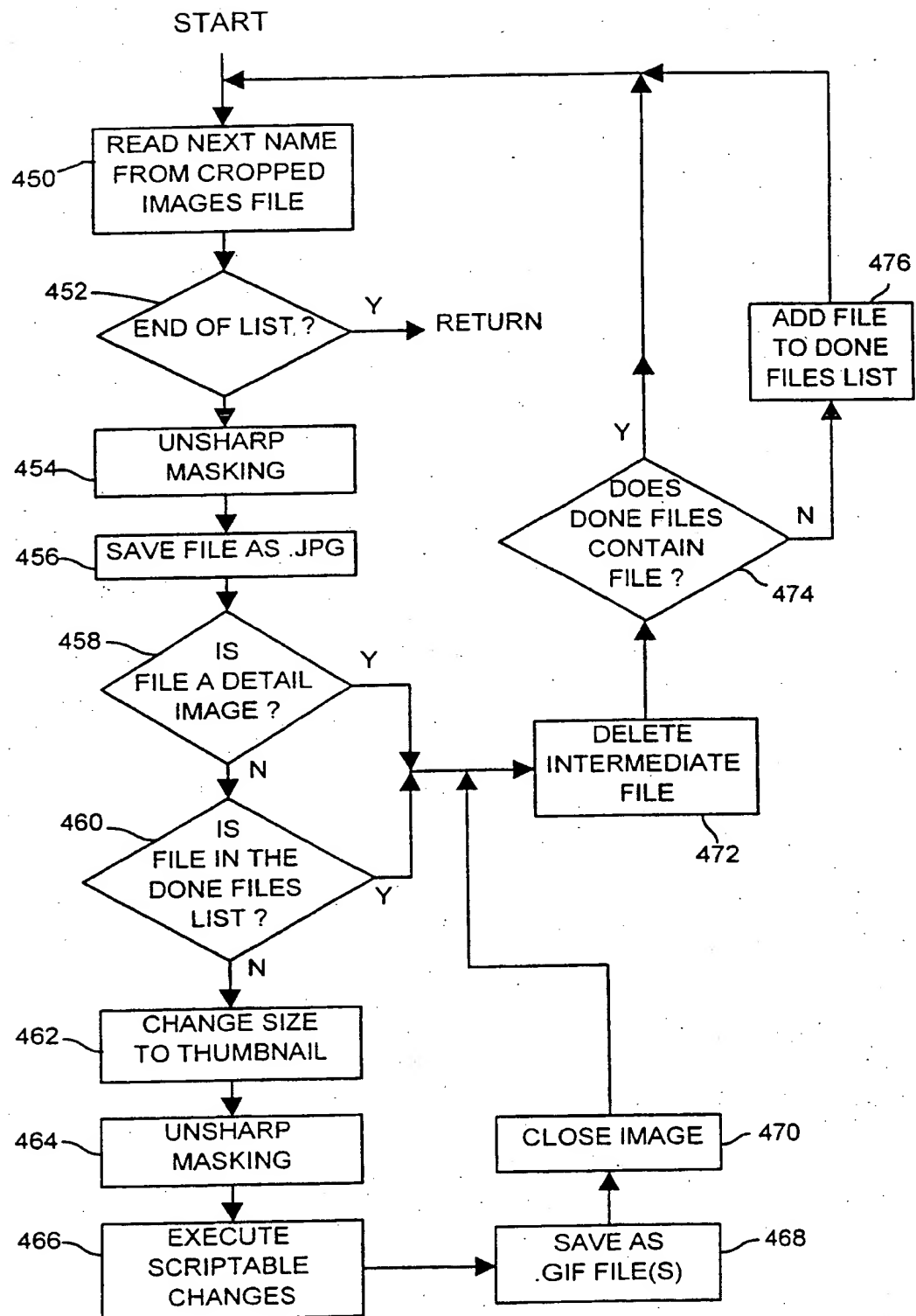


FIG. 22

25 / 26

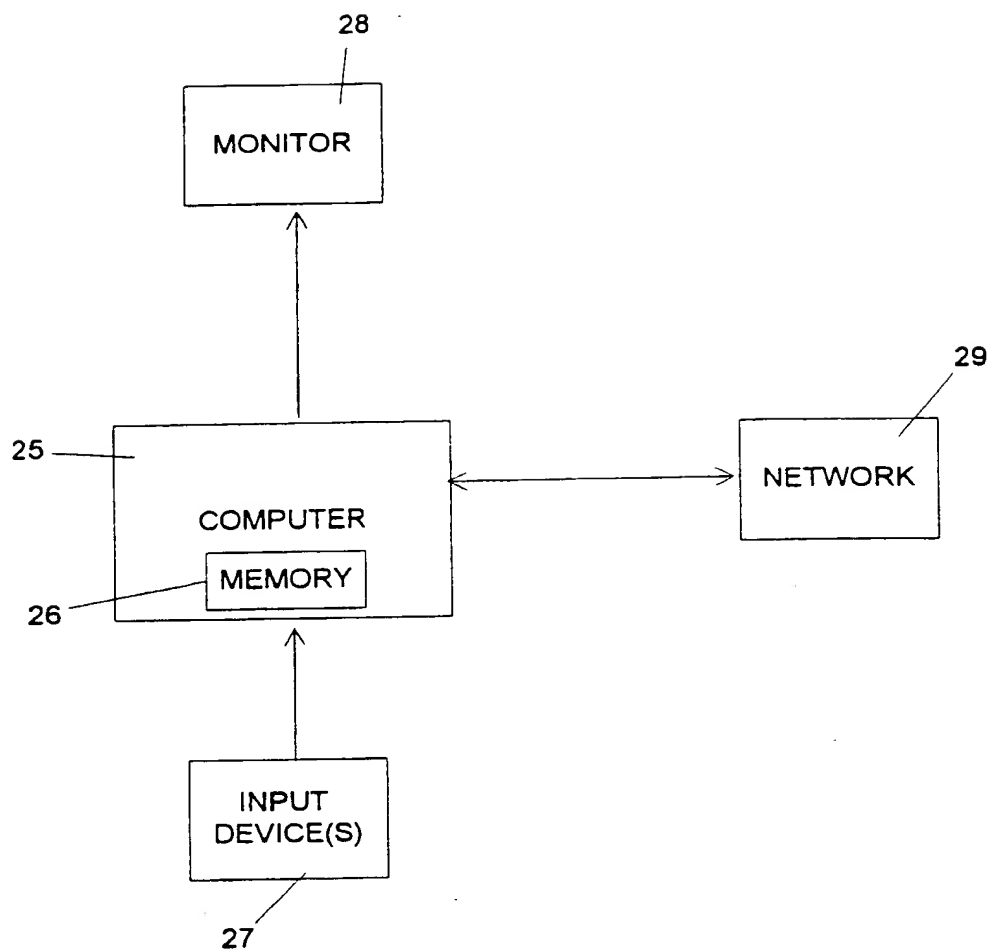


FIG. 23

26 / 26

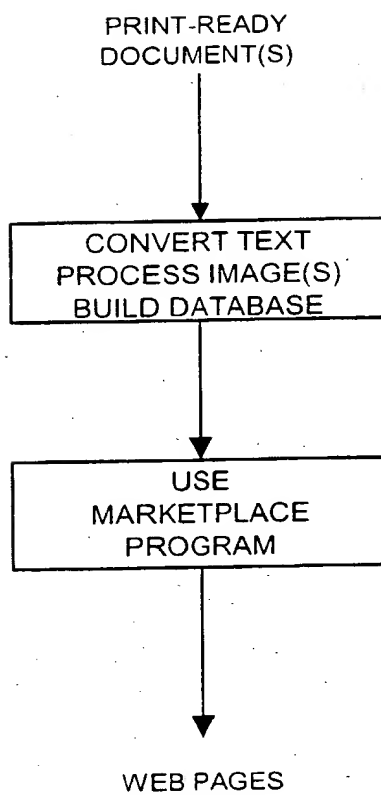


FIG. 24

# INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 99/20203

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/24 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MASAAKI MIZUNO ET AL: "DOCUMENT RECOGNITION SYSTEM WITH LAYOUT STRUCTURE GENERATOR" NEC RESEARCH AND DEVELOPMENT, JP, NIPPON ELECTRIC LTD. TOKYO, vol. 32, no. 3, 1 July 1991 (1991-07-01), pages 430-437, XP000265886 ISSN: 0547-051X abstract page 430, column 2, line 10 - line 14 page 432, column 1, line 15 - line 17 --- -/--	1-4, 10-13, 28-31, 38-41

☒ Further documents are listed in the continuation of box C.

☐ Patent family members are listed in annex.

### \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

\*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

\*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

\*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

\*G\* document member of the same patent family

Date of the actual completion of the international search

31 May 2000

Date of mailing of the international search report

07/06/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

González Arias, P

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 99/20203

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>ROUSSEAU B ET AL: "WRITING DOCUMENTS FOR PAPER AND WWW A STRATEGY BASED ON FRAMEMAKER AND WEBMAKER" COMPUTER NETWORKS AND ISDN SYSTEMS, NL, NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 27, 1 January 1994 (1994-01-01), pages 205-214, XP000571730 ISSN: 0169-7552 abstract page 208, column 1, line 31 - line 46</p>	1-46
A	<p>PATENT ABSTRACTS OF JAPAN vol. 1999, no. 13, 30 November 1999 (1999-11-30) &amp; JP 11 238053 A (MATSUSHITA ELECTRIC IND CO LTD), 31 August 1999 (1999-08-31) abstract</p>	1-46

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 99/20203

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 11238053 A	31-08-1999	NONE	

**THIS PAGE BLANK (USPTO)**